ABC INFORMATIQUE

N° 68

COURS D'INFORMATIQUE PRATIQUE ET FAMILIALE

L'avenir des robots

Essayez le Compaq Plus

Diagrammes par logo

Informatique sans frontière

EDITIONS ATLAS

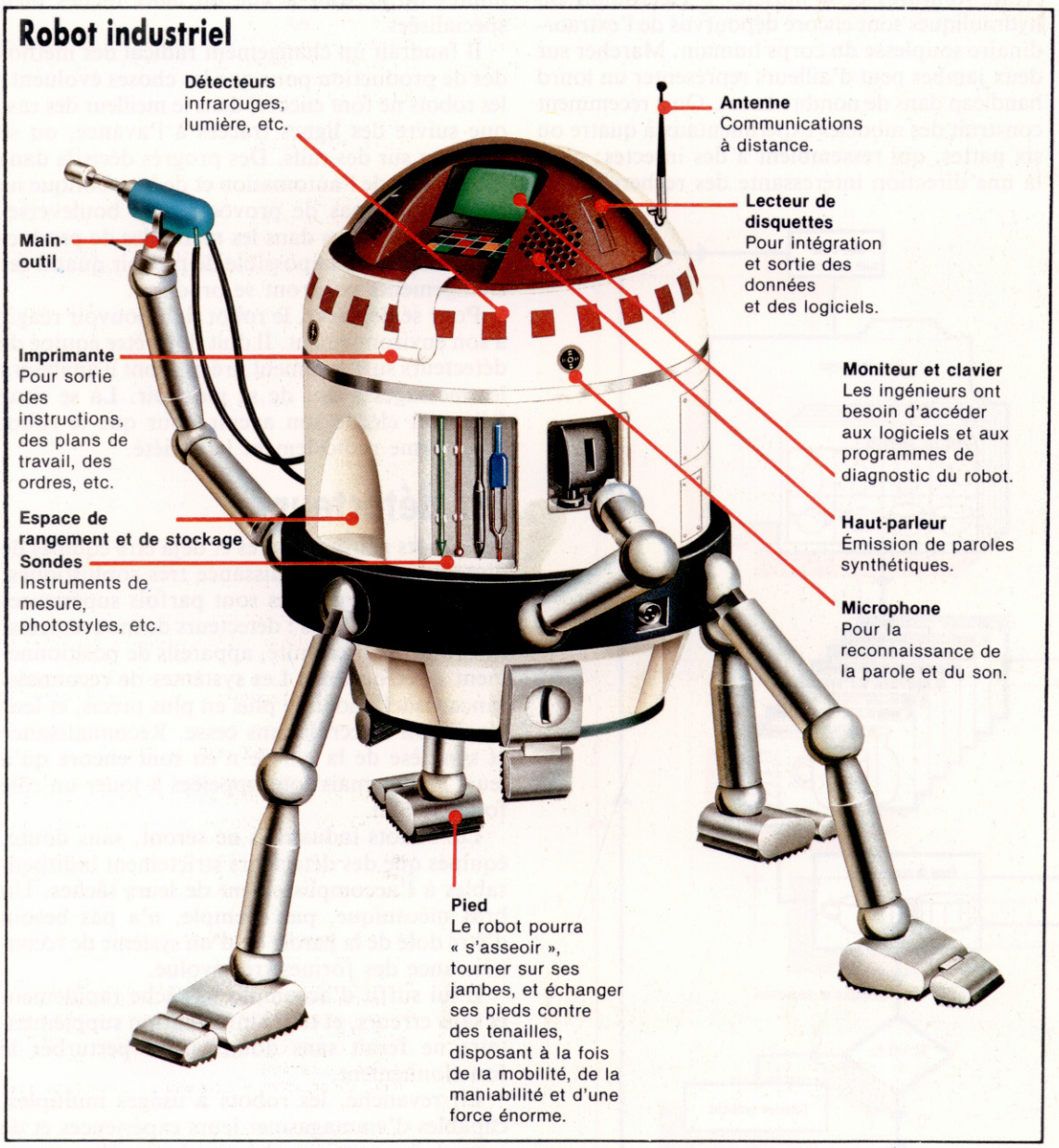
M 6062-68-12,50 F 90 FB - 3,80 FS - $ 1.95

Page manquante
(publicité et colophon)

Les robots et leur avenir

Jusqu'ici, nous avons passé en revue tous les aspects du comportement d'un robot. Pour conclure, voyons quelles sont les limites des modèles actuels, et quels progrès on peut attendre.

Robot industriel



L'avenir
Si jamais on parvient à mettre au point, pour un coût raisonnable, un robot humanoïde capable de se substituer aux humains pour certaines tâches semi-spécialisées, il devra disposer d'une « intelligence » hautement évoluée, qui fasse appel à une base de données complexe (connaissances, capacités de travail), à des détecteurs très sensibles, ainsi qu'à un logiciel de commande qui lui permette réellement d'apprendre. Un tel modèle pourrait être installé dans des « carrosseries » très différentes; nous avons représenté ici un exemple possible.
(Cl. Steve Cross.)

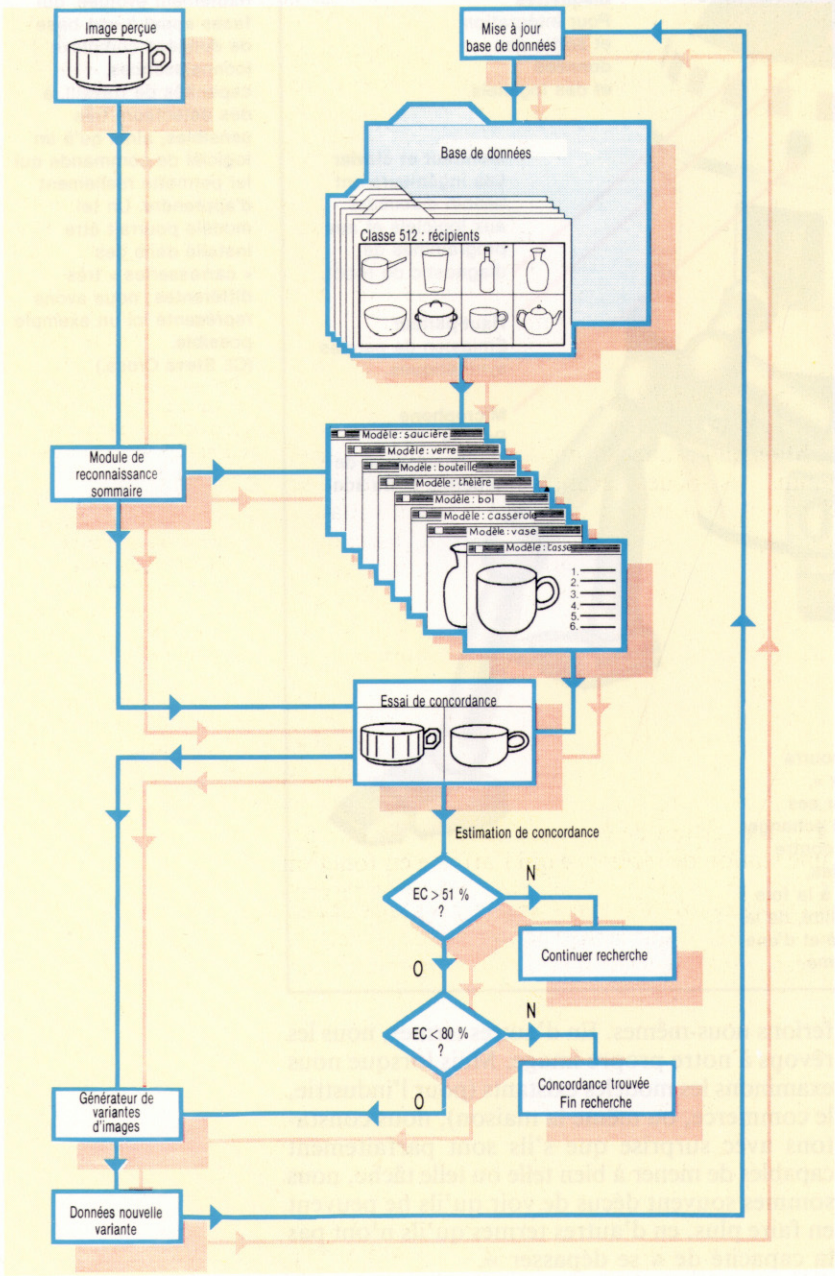
Les robots d'aujourd'hui sont encore bien loin des êtres mécaniques pensants qu'on imagine souvent. Nous nous attendons à retrouver chez eux certaines qualités : pouvoir se déplacer librement, de leur propre initiative; voir et entendre le monde qui les entoure; discuter avec nous de problèmes philosophiques et scientifiques, ou du moins communiquer de façon intelligente; manipuler des objets et des idées comme nous le ferions nous-mêmes. En d'autres termes, nous les rêvons à notre propre image. Mais lorsque nous examinons les modèles existants (pour l'industrie, le commerce, ou même la maison), nous constatons avec surprise que s'ils sont parfaitement capables de mener à bien telle ou telle tâche, nous sommes souvent déçus de voir qu'ils ne peuvent en faire plus, en d'autres termes qu'ils n'ont pas la capacité de « se dépasser ».

Thème et variations
Pour reconnaître un objet, un robot peut garder en mémoire un « modèle » de base, auquel viennent se superposer des « variantes ». Les détecteurs de l'engin fourniront une image analysée sommairement par un module de reconnaissance, ce dernier émettant un premier pronostic sur la classe à laquelle appartient l'objet considéré. Chacune des images de cette classe sera alors confrontée avec l'objet de référence, jusqu'à ce qu'on trouve une concordance après analyse statistique. (Cl. Ian McKinnell.)

A partir de ce que l'on sait déjà, cherchons à préciser ce que l'on peut, concrètement, attendre des robots de l'avenir en considérant, séparément, leurs caractéristiques essentielles.

Le mouvement

Il est très improbable que les robots puissent, dans un avenir proche, se déplacer sur des supports ressemblant à une jambe humaine. Maintenir l'appareil en équilibre demanderait trop de temps de traitement et d'espace mémoire. De surcroît, jointures et « muscles » électriques ou hydrauliques sont encore dépourvus de l'extraordinaire souplesse du corps humain. Marcher sur deux jambes peut d'ailleurs représenter un lourd handicap dans de nombreux cas. On a récemment construit des modèles expérimentaux à quatre ou six pattes, qui ressemblent à des insectes; c'est là une direction intéressante des recherches.



Pour d'autres applications, comme les opérations militaires, l'exploration des planètes ou les simples usages domestiques, on continuera sans doute à recourir aux roues. C'est le système le plus pratique. Le déplacement des robots sera certainement plus fluide, mais sans pour autant égaler, bien sûr, l'aisance et la puissance d'un athlète en mouvement.

Le montage à la chaîne, tel qu'il existe actuellement, implique d'ailleurs l'immobilité des robots industriels et des bras mécaniques, qui ne peuvent se déplacer que sur un espace restreint, limités qu'ils sont à une ou deux tâches bien spécialisées.

Il faudrait un changement radical des méthodes de production pour que les choses évoluent; les robots ne font encore, dans le meilleur des cas, que suivre des lignes tracées à l'avance, ou se déplacer sur des rails. Des progrès décisifs dans le domaine de l'automation et de la robotique ne manqueront pas de provoquer les bouleversements nécessaires dans les méthodes de production, mais il est impossible de prévoir quand ces changements pourront se produire.

Pour se déplacer, le robot doit pouvoir réagir à son environnement. Il doit donc être équipé de détecteurs suffisamment précis, dont il analysera les messages avant de se mouvoir. Là se situe l'élément clé de son avenir pour que le robot transforme radicalement la société.

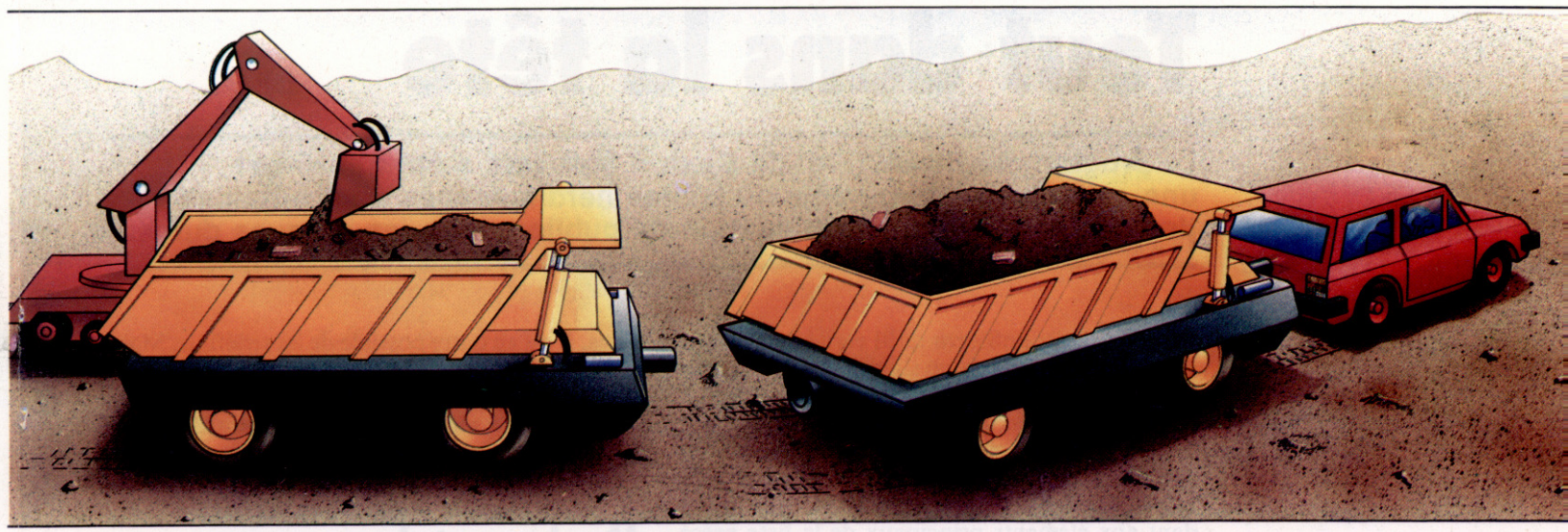
Les détecteurs

Les robots peuvent d'ores et déjà être équipés de dispositifs de reconnaissance très sophistiqués, dont les performances sont parfois supérieures à celles des humains : détecteurs de mouvements, de bruits, de proximité, appareils de positionnement à rétroaction. Les systèmes de reconnaissance visuelle sont de plus en plus précis, et leur résolution s'accroît sans cesse. Reconnaissance et synthèse de la parole n'en sont encore qu'à leurs débuts mais sont appelées à jouer un rôle fondamental.

Les robots industriels ne seront, sans doute, équipés que des détecteurs strictement indispensables à l'accomplissement de leurs tâches. Un bras mécanique, par exemple, n'a pas besoin d'être doté de la parole ou d'un système de reconnaissance des formes très évolué.

Il lui suffit d'accomplir sa tâche rapidement et sans erreurs, et toute information supplémentaire ne ferait sans doute qu'en perturber le fonctionnement.

En revanche, les robots à usages multiples, capables d'emmagasiner leurs expériences et de reproduire certains schémas de pensée proprement humains devront être pourvus de tous les détecteurs possibles. Ils seront capables d'analyser leur environnement, de rassembler des informations et, surtout, d'en faire usage. On sait que les êtres humains ne comprennent pas la parole uniquement à l'aide de l'ouïe : ils se fient, en réalité, à une combinaison de stimuli sonores et visuels. Cet aspect synthétique de la communication, souvent négligé lors de la construction de



robots, est fondamental si l'on veut qu'ils comprennent réellement un jour le langage et puissent correspondre réellement avec les êtres humains.

Autre problème : un robot ne peut traiter à la fois qu'un nombre de données limité, pour des raisons d'espace mémoire et parce que le traitement des données prend beaucoup de temps. Il lui est possible de stocker l'image d'un objet — une pomme, par exemple — et de la faire correspondre à un nom. Mais conserver une image occupe de la place en mémoire, surtout si sa résolution est élevée. De plus, toutes les pommes ne se ressemblent pas; il faudra donc soit plusieurs images de pommes différentes, soit un algorithme capable de reconnaître diverses variantes ou de manipuler l'image de base de telle façon qu'elle soit visible sous n'importe quel angle. Même avec une résolution minimale (disons 256 pixels par image), le nombre de variantes peut dépasser le millier.

Les problèmes d'espace mémoire pourront sans doute être surmontés grâce à des puces RAM plus importantes (on met actuellement au point des modèles de 1 méga-octet), ainsi qu'à des puces spécialisées capables de stocker des données « variables », qui seront ensuite appelées par le microprocesseur, un peu à la manière d'un GOSUB en BASIC.

Outre un espace mémoire colossal, il faudra aussi au robot des sources d'informations aussi variées que possible dont les données seront traitées en un minimum de temps. C'est une tâche dont les microprocesseurs actuels ne peuvent s'acquitter : ils sont trop lents. Sans doute faudra-t-il faire travailler en parallèle deux unités centrales (ou plus) à grande vitesse et haute capacité, tandis qu'un processeur de contrôle distribuerait les tâches à d'autres dispositifs implantés dans le système.

De ce point de vue, les progrès sont extrêmement rapides. Reste à fournir au robot un logiciel suffisamment complexe pour qu'il puisse comprendre le monde qui l'entoure, et les besognes qu'il doit accomplir. En d'autres termes, il faut lui apprendre à penser.

La pensée

Comme nous l'avons déjà vu, deux grandes voies s'offrent à la robotique. La première, dans laquelle elle s'est déjà engagée, est celle des outils intelligents (bras mécaniques, chaînes de montage automatisées, etc.). Ils sont déjà capables de mener à bien des tâches très complexes, pour peu qu'on les dote de logiciels de contrôle d'une précision suffisante. Un bras mécanique a besoin d'un ensemble de coordonnées et d'une séquence d'actions, il n'est pas nécessaire qu'il comprenne quoi que ce soit. Il pourra ainsi, sans la moindre erreur, peindre une porte de voiture sur une chaîne de montage, ou laver la même voiture dans un garage.

Mais plus il aura de choses différentes à faire, moins ces tâches seront précises : s'il doit se déplacer dans une pièce dont le contenu change chaque jour, il devra non seulement rassembler et traiter des informations, mais surtout les intégrer dans leur propre conception du monde qui les entoure. Le logiciel de contrôle devra dans ce cas lui permettre une certaine souplesse de manœuvre.

Chercher à créer un cerveau « mécanique » soulève des questions très importantes, encore non résolues sur le propre mécanisme du cerveau humain. Quelles sont par exemple les « connaissances » dont dispose un nouveau-né? Un adulte est-il entièrement le produit de son hérédité ou de son environnement? Quelle est l'interaction entre les deux? Un être humain possède-t-il dès le « départ » une construction particulière, un « réseau interne » qui lui permet d'apprendre une langue, les mathématiques, d'élaborer une philosophie? Si tel est le cas, comment cela fonctionne-t-il? Difficile de répondre, d'autant que les expériences directes sur le cerveau humain sont très délicates! Peut-être que dans le futur, les robots qui seront créés à notre image nous aiderons à nous comprendre nous-mêmes. Mais si de telles questions ne sont pas que pures théories — les expériences en cours en donnent la preuve —, l'idée de voir un jour un robot « pensant » naître ne se concrétisera pas demain!

En route!
La robotique jouera sans doute son plus grand rôle dans la mise au point de véhicules spécialisés : grues, pelleteuses, transports, livraisons à domicile. Nous avons représenté ici un robot-pelleteuse remplissant un convoi de camions-robots sur un chantier de construction. Une fois chargé, chaque camion se dirige, de façon semi-intelligente, vers un point de rassemblement, et vient s'intégrer à un convoi d'autres camions qui emprunte alors les routes, sous le contrôle étroit du chauffeur (humain) du véhicule de tête. Soumettre la force brute et l'intelligence limitée des robots aux pouvoirs de contrôle et de décision de l'homme constituera sans aucun doute le meilleur moyen de mettre en œuvre toutes les ressources offertes par les nouvelles technologies.
(Cl. Kevin Jones.)

Tout dans la tête

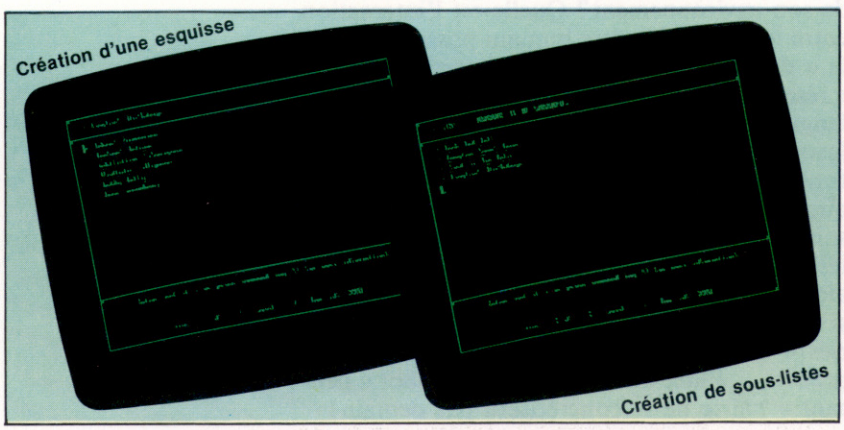
Le programme BrainStorm de Caxton Software a été utilisé par une association de parents de jeunes héroïnomanes pour organiser des campagnes de sensibilisation du public contre les drogues dures.

On a dit de BrainStorm qu'il était un « processeur de pensée »; cela ne signifie pas que le programme a pour but de se substituer à l'homme. Il se contente de l'aider à organiser ses pensées.

Il vous aide à dresser les listes des étapes à suivre lorsque vous planifiez un projet. La liste des principaux objectifs crée des sous-listes répondant aux éléments principaux de la liste précédente et ainsi de suite. Cette arborescence s'amplifie jusqu'à ce que l'on s'y perde. Toute modification — décider qu'un élément de la liste principale doit passer dans une sous-liste d'un autre élément par exemple — peut supposer de telles ratures et corrections que le système deviendra inutilisable. Avec BrainStorm, c'est aussi facile à faire que d'utiliser les fonctions de déplacement de paragraphes pour le traitement de texte. C'est pourquoi l'expression « processeur de pensée » n'est pas si fausse.

Chaque élément d'une liste ou d'une sous-liste peut devenir le titre d'une liste de niveau hiérarchique inférieur. Lorsque des activités similaires dans différentes listes doivent être regroupées, elles reçoivent le même nom ou la même rubrique. En l'occurrence, tout ce qui est ajouté à une sous-liste figurant dans une rubrique est également ajouté à toutes les autres listes sous la même rubrique.

Quiconque a connu les commandes d'édition plutôt compliquées d'un progiciel de traitement de texte comme WordStar trouvera très simples les commandes non mnémoniques de contrôle du curseur. La frappe simultanée des touches CONTRÔLE et G déplace le curseur vers la gauche. CTRL (ConTRôLe) — D, CTRL — H et CTRL — B déplacent respectivement le curseur à droite, en haut et en bas; les quatre lettres dessinant une rosace sur le clavier qui correspondent aux quatre directions. Il est en outre possible de les redéfinir. Mais il n'est pas évident que pour monter ou descendre dans le texte à l'aide du curseur, il faille passer en mode « modification » (en faisant CTRL — M), pour se déplacer sur une même ligne vers la droite ou vers la gauche.



Création d'une esquisse

Création de sous-listes

Maniement du programme

Le menu principal de BrainStorm présente onze fonctions, la sélection d'une fonction se faisant par une lettre qui est souvent très explicite : Utiliser, Charger, Afficher, Identité Lecteur (pour affecter un autre lecteur que celui par défaut), Effacer (pour effacer le modèle courant de la mémoire), Sauvegarder, Écrire (pour recopier sur disque), Répertoire, sOrtie, Fusion et Détruire.

Pour commencer, l'utilisateur appuie sur Départ, le programme passant presque instantanément en mode saisie. Les idées sont saisies sous forme d'une liste plus ou moins organisée (appelée « modèle »). Voici un exemple de modèle :

Lire manuel
Taper la liste
Taper la sous-liste
Éditer la liste

Les commandes de contrôle n'apparaissent pas immédiatement, mais sont listées en tapant. En déplaçant le curseur vers le haut, sur un élément de la liste, et en faisant CTRL — T, l'élément ainsi désigné devient le titre d'une sous-liste. De manière similaire, chaque élément d'une sous-liste peut devenir le titre d'une nouvelle sous-liste de niveau inférieur, et ainsi de suite jusqu'à l'épuisement de la mémoire disponible. Pour revenir à la liste souche, faire CTRL — S.

Les éléments peuvent être déplacés à l'intérieur d'une liste, ou d'une liste à l'autre. Pour cela, il suffit de leur attribuer le signe @, et de faire ensuite CTRL — R pour Retirer, ou CTRL – I pour Insérer. Une fois effectué CTRL – R, le signe @ se place automatiquement sur le prochain élément de la liste, et ainsi de suite. En faisant à nouveau CTRL — R, cet élément est pris à son tour. Cette caractéristique se révèle utile pour déplacer une liste entière d'éléments sur une autre sous-liste.

Si vous désirez insérer de nouveaux éléments dans une liste, il vous faut simplement vous positionner sur le début de la ligne devant suivre l'insertion, et la taper. Lorsque vous validez cette saisie, le reste de la liste est décalé d'une ligne. CTRL — M permet de modifier un élément. On peut, en donnant aux éléments de plusieurs listes

le même nom, créer une rubrique commune à plusieurs listes. Si une liste donnée supposait qu'un événement particulier se produise un certain jour, mettons le 1er janvier, il est possible de créer une liste des événements pour cette date. Cela permet de tenir à jour l'emploi du temps de cette journée par des recoupements.

Il est possible de créer autant de rubriques que l'on veut et d'y accéder l'une après l'autre, avec CTRL — A et CTRL — P, pour obtenir respectivement la rubrique d'Après et la rubrique Précédente. Les rubriques sont en boucle sans fin, CTRL – A redonnant après la dernière rubrique la première de la séquence.

Affichage de listes

Il est bien sûr possible d'afficher les listes, leur format faisant apparaître des renfoncements de ligne pour signifier leur niveau. Par exemple :

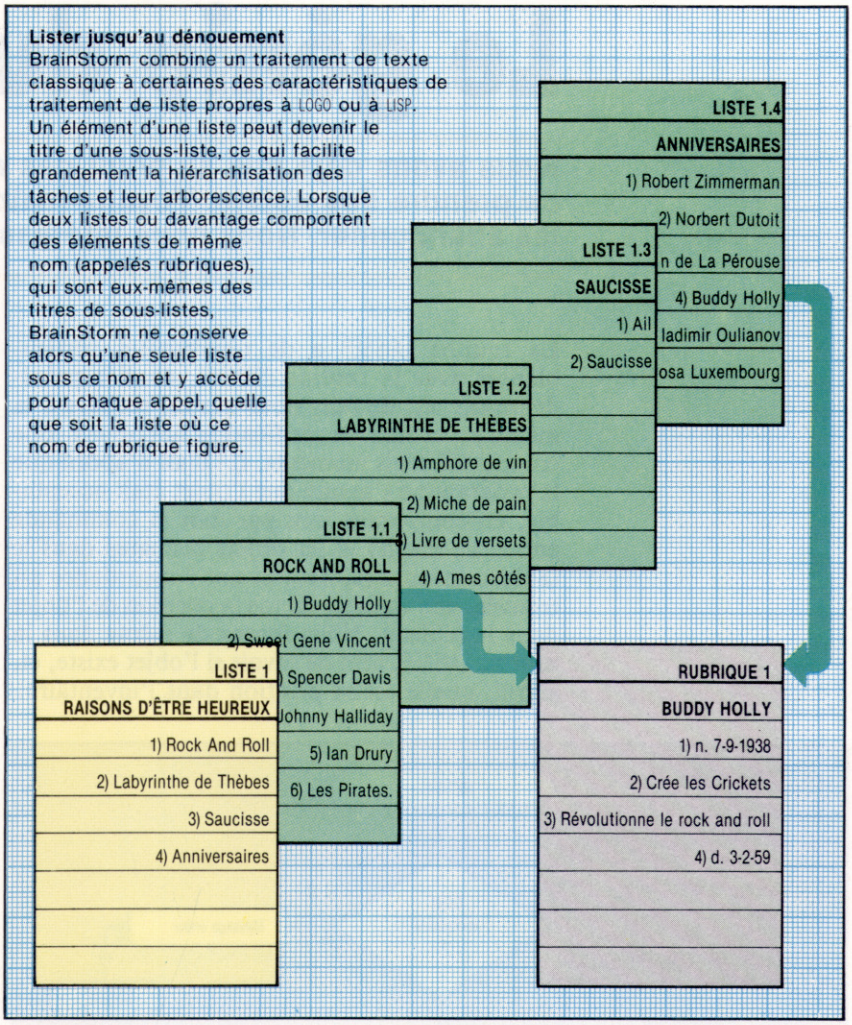
Lire le manuel
  Prendre le manuel dans son étui
    Aller à la table des matières
      Trouver la bonne page
    Lire la page
  Remettre le manuel dans son étui
Taper la liste
  Taper CTRL – T pour faire d'un élément de la liste un nouveau titre
    Taper la sous-liste
  Faire CTRL – P pour revenir à la liste précédente
Éditer la liste
  Faire CTRL – M pour modifier une saisie

Le nombre de renfoncements est décidé par l'utilisateur.

Il est également possible d'éditer des listes de manière externe à BrainStorm. Si elles ont été sauvegardées par la commande Écrire-sur-disque, elles seront reconnues comme documents par divers traitements de texte. Elles pourront être éditées, affichées, et, si nécessaire sauvegardées à nouveau. BrainStorm peut donc être utilisé, par exemple, pour établir le synopsis d'un livre et, ensuite, l'écrire en détail à l'aide d'un traitement de texte traditionnel. Ce dernier a alors accès au fichier BrainStorm.Doc pour développer le livre.

Les premières idées, les grandes lignes du projet, deviennent des titres de chapitre (qui contiennent chacun une liste spécifique). Lorsqu'un élément d'un chapitre (une liste) prend suffisamment d'importance pour devenir à lui seul un chapitre, il devient effectivement un nouveau titre de liste. Réciproquement, des titres de chapitres peuvent se révéler de moindre importance que ce qui avait été escompté : ils peuvent alors devenir de simples éléments de liste.

L'organisation générale du livre peut ainsi prendre forme, et, comme WordStar ou d'autres traitements de texte peuvent accéder aux fichiers de BrainStorm, le développement de cette ébauche jusqu'au manuscrit final va en découler de manière naturelle. Cela signifie qu'il n'est pas nécessaire de passer par le formatage d'impression propre à BrainStorm, qui consiste à décider d'un double ou d'un triple espacement...



Lister jusqu'au dénouement
BrainStorm combine un traitement de texte classique à certaines des caractéristiques de traitement de liste propres à LOGO ou à USP. Un élément d'une liste peut devenir le titre d'une sous-liste, ce qui facilite grandement la hiérarchisation des tâches et leur arborescence. Lorsque deux listes ou davantage comportent des éléments de même nom (appelés rubriques), qui sont eux-mêmes des titres de sous-listes, BrainStorm ne conserve alors qu'une seule liste sous ce nom et y accède pour chaque appel, quelle que soit la liste où ce nom de rubrique figure.

Le programme est livré avec un modèle-type (appelé MODÈLE.BRN), qui comprend la trame d'un système d'agenda avec fichier « nom-et-adresse », une liste des tâches (dont les sous-listes portent les titres « urgent », « important » et « ne pas oublier »), plus une partie bloc-notes. Ce modèle, intéressant, peut être ajouté à ses propres modèles à l'aide de l'option « fusion ».

BrainStorm est d'un apprentissage facile. Le classeur-manuel est très clair (lui-même est écrit avec BrainStorm). Les commandes sont toujours accessibles depuis un menu d'écran, ce qui rend superflu le recours systématique au manuel.

Toutes les commandes ne sont pas mnémoniques, aussi certaines seront difficiles à retenir pour l'utilisateur; mais il existe un programme appelé INSTALLB qui permet de reconfigurer les commandes et de modifier les menus en conséquence. Ce programme est extrêmement clair et entièrement piloté par menu.

BRAINSTORM : Concerne quelque 25 machines CP/M, MS-DOS et PC-DOS, dont l'IBM, le Sirius et l'Apricot.

DIFFUSION : Caxton Software.

AUTEURS : David Tebbut et Mike Liardet.

FORMAT : Disque.

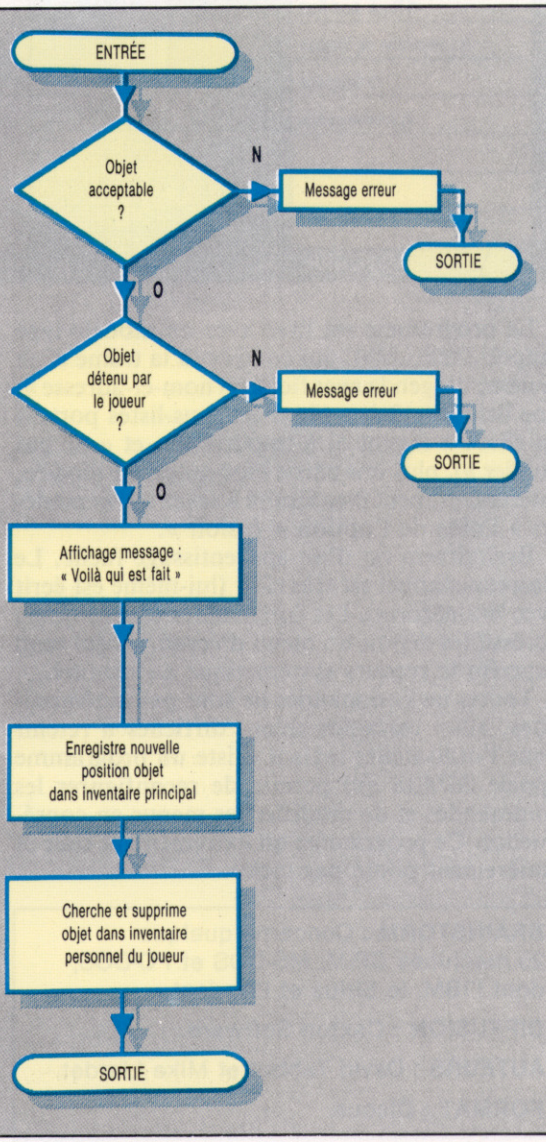
Le génie du lieu

Nous avons déjà mis au point des routines permettant au joueur de prendre des objets. Voyons maintenant de quelle façon il peut les abandonner. Nous nous intéresserons aussi aux « Lieux spéciaux ».

La routine LAISSER a de nombreuses ressemblances avec la routine PRENDRE, étudiée précédemment, et, de fait, nous aurons recours à la même procédure de vérification des objets. Elle comportait trois moments. Il fallait d'abord que le programme s'assure que l'objet était acceptable, en comparant mot par mot la commande entrée au clavier et les objets contenus dans le tableau-inventaire, IV$(,). Si une concordance est trouvée, une variable donne la position de l'objet dans le tableau. Nous ferons de même dans la routine LAISSER, afin de savoir si l'objet existe, et, si oui, quelle est sa position dans l'inventaire.

Laissez tomber!

La routine LAISSER suit les mêmes règles que la routine PRENDRE, mais il lui suffit de constater que l'objet est acceptable et qu'il est bien aux mains du joueur, pour qu'elle soit exécutée.
L'inventaire principal est alors mis à jour pour enregistrer le nouvel emplacement de l'objet (et le nom de l'objet est supprimé de l'inventaire des possessions du joueur).
(Cl. Ian McKinnell.)



Ensuite, la routine vérifiait si, oui ou non, le joueur était déjà en possession de l'objet en analysant le tableau des objets transportés, IC$(). Nous ferons de même ici : de toute évidence, il est impossible d'abandonner quelque chose qu'on n'a pas sur soi! Pour finir, la routine s'assurait que l'objet spécifié se trouvait bien dans le lieu occupé par le joueur, déterminé par la variable de position P. Mais cette fois-ci, ce test est inutile, puisque l'objet est déjà transporté. Nous n'en ferons donc pas usage.

Cela étant posé, et si les deux tests de vérification donnent des résultats positifs, nous aurons besoin de procéder à deux mises à jour dans l'inventaire principal et dans celui du joueur :

1. La position de l'objet abandonné sera désormais spécifiée par la variable F. La position présente, P, sera intégrée au tableau principal en IV$(F,2).
2. La description de l'objet sera supprimée de l'inventaire des objets transportés, IC$(). Le meilleur moyen, pour ce faire, est encore de chercher dans le tableau, jusqu'à ce que l'objet en question soit trouvé; on la remplacera alors par une chaîne vide.

L'organigramme ci-contre explique la logique de la routine LAISSER. En voici le listage pour la Forêt hantée :

```
3900 REM **** S/P LAISSER ****
3910 GOSUB5300:REM OBJET ACCEPTABLE
3920 IF F=0 THEN SN$= "JE N'EN VOIS PAS" GOSUB5500:
RETURN
3930 :
3940 REM ** L'OBJET EST-IL TRANSPORTE **
3950 OV=F:GOSUB5450
3960 IF HF=0 THEN SN$ "VOUS N'EN AVEZ PAS"
:GOSUB5500:RETURN
3970 :
3980 REM ** LAISSER OBJET **
3990 SN$= "VOILA QUI EST FAIT":GOSUB5500
4000 IV$(F,2)=STR$(P):REM ENTREE DANS INVENTAIRE
4010 :
4020 REM ** SUPPRIME OBJET DANS INVENTAIRE OBJETS
TRANSPORTES **
4030 FOR J=1TO2
4040 IF IC$(J)=IV$(F,1) THEN IC$(J)="":J=2
4050 NEXT J
4060 RETURN
```

On voit que le principal avantage d'une programmation de type modulaire est qu'une même routine peut servir à plusieurs tâches différentes. L'emploi d'une série de drapeaux permet de prendre très rapidement des décisions grâce à des sous-routines qui n'ont pas de conséquences avant que le contrôle ne soit repassé à la routine d'appel. Le test de validité décrit précédemment est un bon exemple d'emploi de ce procédé. Il est appelé aussi bien par PRENDRE que par LAISSER; dans les deux cas, il permet de savoir si oui ou

non l'objet spécifié est acceptable. Cependant, le déroulement du programme n'en est pas affecté avant la fin de la routine, marquée par un RETURN. C'est seulement lorsqu'il reçoit la valeur de la variable F qu'il effectue la mise à jour nécessaire. L'un des inconvénients de cette méthode est qu'elle teste deux fois de suite la même condition : une fois pour définir la valeur du drapeau, une fois pour la tester. L'objection est fondée, mais, si le temps d'exécution est légèrement plus long, le procédé offre plus de souplesse et surtout reste plus facile à déboguer.

« Lieux spéciaux »

Nous sommes maintenant en possession du « squelette » de notre jeu; le joueur peut désormais se déplacer dans notre univers imaginaire, et y prendre des objets. Reste à passer à la phase suivante, celle du traitement des « lieux spéciaux » : c'est là que les objets sont utilisés, et que l'ingéniosité et l'habileté du joueur peuvent donner toute leur mesure face aux périls.

Avant d'examiner en détail la programmation des routines consacrées à l'un des lieux de la Forêt hantée, voyons d'abord quelles sont les modifications à apporter au programme principal pour qu'il soit en mesure de détecter des endroits de ce type. Il suffit d'insérer ces deux lignes :

```
257 GOSUB 2700 : REM P EST-IL UN ENDROIT SPECIAL ?
258 IF SF=1 THEN 300 : REM INSTRUCTION SUIVANTE
```

La ligne 257 renvoie à une routine pour voir si le lieu occupé par le joueur présente quelque chose de spécial. Si c'est le cas, un drapeau particulier, SF, se voit affecter une valeur de 1. Lorsque le programme revient à la boucle principale, tout ce qui est relatif aux instructions peut ainsi être laissé de côté. La sous-routine qui en décide se présente ainsi :

```
2700 REM **** S/P P EST-IL SPECIAL ? ****
2705 SF=0 : REM LE DRAPEAU EST MIS A ZERO

2716 REM **** AUTRES LIEUX SPECIAUX ****
2720 ON P GOSUB 4590,4690,4790,4590
2730 RETURN
```

Vous vous souvenez certainement que lorsque nous avons tracé la carte de la Forêt hantée, nous avons numéroté en premier les « lieux spéciaux ». Nous pouvons ainsi simplifier la sélection de chacune des sous-routines qui leur est consacrée, par simple usage de la commande ON...GOSUB. Comme on le voit à la ligne 2720, elle est suivie d'une série de numéros de ligne, l'un d'entre eux étant choisi selon la valeur de P. Si cette valeur est égale à 1, par exemple, le GOSUB renverra au premier numéro de la liste; si elle est égale à 2, au deuxième, et ainsi de suite.

Il y a en tout quatre numéros de lignes différents, un pour chaque emplacement. Si P est supérieur à 4, le programme passe tout simplement à la ligne suivante. Si chacune des quatre sous-routines appelées à partir de la ligne 2720 met en place un drapeau SF, il s'en suit que si P correspond à un « lieu spécial », cela peut être indiqué de la même façon. S'il n'y a pas d'appel de sous-routine, SF reste égal à zéro, indiquant par là que P est un endroit sans signification particulière. ON...GOSUB est bien plus pratique qu'une série d'instructions IF...THEN, qui testent la valeur d'une variable et, en fonction du résultat, appellent telle ou telle sous-routine.

L'entrée du tunnel

Les deux entrées d'un tunnel (lieux 1 et 4) constituent deux des « lieux spéciaux » de la Forêt hantée. Pour que le joueur puisse pénétrer dans le tunnel, nous devons mettre au point très soigneusement une routine qui lui permette à la fois d'entrer et de battre en retraite vers le chemin.

```
4590 REM **** S/P ENTREE DU TUNNEL ****
4600 SF=1
4605 SN$="VOUS VOILA A L'ENTREE D'UN VASTE
TUNNEL " : GOSUB 5500
4610 SN$="VOUS POUVEZ Y PENETRER OU RECULER"
: GOSUB 5500
4620 :
4625 PRINT : INPUT "INSTRUCTIONS": IS$
4630 GOSUB 2500 : REM DECOMPOSE COMMANDE
4635 IF F=0 THEN 4625 : REM COMMANDE NON VALABLE
4637 GOSUB 3000 : REM PAS DE PROBLEMES
4640 IF MF=1 THEN RETURN : REM LE JOUEUR BAT
EN RETRAITE
4645 IF VF=1 THEN 4625 : REM LE PROGRAMME
OBEIT
4650 REM **** NOUVELLES INSTRUCTIONS ****
4655 IF VB$="ENTRER" THEN GOSUB 4700 : RETURN
4660 IF VB$="RECULER" AND P=4 THEN MF=1 :
P=6 : RETURN
4665 IF VB$="RECULER" AND P=1 THEN MF=1 :
P=9 : RETURN
4667 SN$=" JE NE COMPRENDS PAS" : GOSUB 5500
: RETURN : GOTO 4625
```

La routine commence en donnant à SF la valeur 1, pour indiquer que l'on vient d'atteindre un lieu spécial. Un message est affiché à l'écran, qui donne une brève description de l'entrée du tunnel et propose deux options différentes au joueur; il lui est alors demandé de faire son choix. Une fois encore, plutôt que de réinventer à chaque fois le processus que nous désirons pour analyser une instruction, nous tirerons parti de la conception modulaire de notre programme en appelant des sous-routines présentes dans les routines PRENDRE et LAISSER.

Il suffit de prendre connaissance de la valeur des différents drapeaux qu'elles contiennent pour mettre en œuvre notre nouveau fragment de code. Voyons ces drapeaux un par un.

F, dans la routine qui décompose la réponse, indique si oui ou non cette dernière est acceptable. S'il s'agit d'un mot isolé que le module ne reconnaît pas, F passe à zéro, ce qui nous renverra en ligne 4625.

MF est activé par la routine « commande normale », au cas où le joueur demande la description d'un endroit (cela arrive avec une commande ALLER ou REGARDER). Un RETURN à la boucle principale permet de se déplacer jusqu'au lieu en question (premier cas), ou bien sa description est affichée de nouveau (second cas).

Entrer et reculer

VF est également mis en œuvre par la routine « commande normale ». S'il est égal à 1, c'est que l'instruction donnée par le joueur a été recon-

nue et exécutée, ce qui nous renvoie à l'instruction suivante. S'il est inférieur ou supérieur à 1, la réponse du joueur ne fait pas partie des commandes acceptables.

Après avoir examiné ces commandes, nous en ajouterons deux : ENTRER et RECULER, qui correspondent aux deux options possibles. Notre routine doit pouvoir fonctionner pour les deux entrées du tunnel, et RECULER devra donc faire la différence entre elles; nous donnerons donc à P soit la valeur 1, soit la valeur 4. P peut ainsi être défini en fonction des circonstances avant la fin de la routine, de sorte qu'un changement de lieu s'effectue dès le retour à la boucle principale du programme.

La prochaine fois, nous traiterons des dangers qui menacent notre aventurier une fois qu'il est entré dans le tunnel. Et ils seront assez nombreux pour ne pas vous ennuyer.

Listages Digitaya

```
1190 GOSUB2670:REM EST-IL SPECIAL
1200 IF SF=1 THEN 1250:REM BOUCLE SUIVANTE

2360 REM ** S/P LAISSER **
2370 GOSUB5730:REM OBJET ACCEPTABLE ?
2380 IF F=0 THEN PRINT"IL N'Y EN A PAS":RETURN
2390 :
2400 REM ** AVEZVOUS L'OBJET ? **
2410 OV=F:GOSUB5830
2420 IFHF=0THEN PRINT "MAIS VOUS NE L'AVEZ PAS !"
:RETURN
2430 :
2440 REM ** LAISSER OBJET **
2450 SN$="VOILA QUI EST FAIT":GOSUB5880
2460 IV$(F,2)=STR$(P):REM UPDATE OBJ POSITION
2470 :
2480 REM ** SUPPRIME OBJET DE LA LISTE **
2490 FORJ=1TO4
2500 IF IC$(J)=IV$(F,1)THENIC$(J)="":J=4
2510 NEXTJ
2520 RETURN

2670 REM **** S/P P SPECIAL ****
2680 SF=0:REM MISE A ZERO DRAPEAU

2710 ON P GOSUB 2850,2960,3450,3830,4180,4550,5150
2720 RETURN

2850 REM **** S/P SORTIE TELE ****
2860 SF=1
2870 SN$="VOUS ETES COINCE DANS LA SORTIE TELE"
2880 SN$=SN$+"VOUS FEREZ PARTIE POUR L'ETERNITE DU
'COLLARO SHOW'."
2890 GOSUB 5880:REM FORMATAGE
2900 PRINT
2910 PRINT "SOUS LE NOM DE PENCASSINE"
2920 FORJ=1TO500:NEXTJ
2930 GOTO 2910
2940 END

3830 REM **** PORT MANCHE A BALAI ****
3840 SF=1
3850 SN$="UN MANIAQUE AUX YEUX ROUGIS TIRE SUR
VOUS AU LASER"

3860 GOSUB5880:REM FORMATAGE
3870 :
3880 REM ** INSTRUCTIONS **
3890 RD=RND(TI):IF RD>.65THEN 4110:REM TOUCHE
3900 PRINT:INPUT"INSTRUCTIONS";IS$
3910 GOSUB1700:GOSUB1900:REM ANALYSE INSTRUCTION
3920 IFMF=1THENMF=0:PRINT"VOUS NE POUVEZ FUIR...":GO
TO3880
3930 IFVF=1THEN3880:REM INSTRUCTION SUIVANTE
3940 IFVB$<>"EMPLOYEZ" THENPRINT "JE NE COMPRENDS PAS":GOTO
3880
3950 GOSUB5730:REM OBJET ACCEPTABLE ?:GOTO3880:REM
 NEXT INSTRUCTION
3970 :
3980 REM ** BOUCLER LASER **
3990 IF F=3 THEN4020:REM OK
4000 SN$= "CA NE SERT A RIEN":GOSUB5880:G
OTO3880
4010 :
4020 OV=3:GOSUB5830:REM BOUCLER LASER TRANSPORTE ?
4030 IFHF=0THENSN$="VOUS N'EN N'AVEZ PAS..."
:GOSUB5880:GOTO3880
4040 :
4050 REM ** SAUVE ! **
4060 SN$="LE BOUCLIER LASER VOUS PROTEGE ET"
4070 SN$=SN$+"UN CHOC VOUS PROJETTE HORS DU PORT
JOYSTICK,VERS LA MACHINE"
4080 GOSUB5880:REM FORMAT
4090 P=INT(RND(TI)*40+7):MF=1:RETURN
4100 :
4110 REM ** HIT **
4120 SN$="TOUCHE PAR LE LASER, VOUS REALISEZ A PEINE"
4140 SN$=SN$+"DANS TOUT L'UNIVERS"
4150 GOSUB5880:REM FORMAT
4160 END


5150 REM **** S/P ENTREE MEMOIRE ****
5160 SF=1
5170 SN$="UN GARDIEN VOUS ACCUEILLE ET POUR VOUS
LAISSER PASSER"
5180 SN$=SN$+"RECLAME UNE ADRESSE":GOSUB58
80
5190 REM ** INSTRUCTIONS **
5200 PRINT:INPUT"INSTRUCTIONS";IS$
5210 GOSUB1700:GOSUB1900:REM ANALYSE
5220 IF MF=1 THEN RETURN:REM RETOUR EN ARRIERE
5230 IF VF=1 THEN 5200:REM NOUVELLE INSTRUCTION
5240 IF VB$<>"DONNE" THENPRINT "JE NE COMPRENDS PAS":G
OTO 5200
5250 :
5260 GOSUB5730:REM OBJET ACCEPTABLE ?
5270 IFF=0THENPRINT "IL N'Y EN A PAS ICI !":GOTO5200:REM
INSTRUCTION SUIVANTE
5280 :
5290 REM ** OBJET:ADRESSE ? **
5300 IF F=1 THEN5330:REM OK
5310 PRINT "IL A BESOIN DE VOTRE ADRESSE":GOTO5200
5320 :
5330 OV=1:GOSUB5830:REM ADRESSE DISPONIBLE ?
5340 IF HF=1 THEN 5370
5350 SN$="VOUS NE L'AVEZ PAS":GOSUB5880:
GOTO5200
5360 :
5370 REM ** ALLEZ-Y **
5380 SN$="LE GARDIEN VERIFIE VOTRE ADRESSE ET"
5390 SN$=SN$+"VOUS LAISSE PASSER":GOSUB5880
5400 P=40:MF=1:RETURN
```

Variantes de basic

Spectrum :

Substituez aux variables originales celles-ci : S$ pour SN$, R$ pour NN$, V$(,) pour IV$(,), I$() pour IC$(), T$ pour IS$, B$ pour VB$.

Dans la Forêt hantée, remplacez les lignes d'origine par celles-ci :

```
2720 IF P=1 THEN GOSUB 4590
2722 IF P=2 THEN GOSUB 4690
2724 IF P=3 THEN GOSUB 4790
2726 IF P=4 THEN GOSUB 4590
```

et dans Digitaya :

```
2710 IF P=1 THEN GOSUB 2850
2711 IF P=2 THEN GOSUB 2960
2712 IF P=3 THEN GOSUB 3450
2713 IF P=4 THEN GOSUB 3830
2714 IF P=5 THEN GOSUB 4180
2715 IF P=6 THEN GOSUB 4550
2716 IF P=7 THEN GOSUB 5159
3890 LET RD=RND(1)
4090 LET P=INT(RND(1)*40+7)
```

Un vrai « pro »

Les micros compatibles IBM se multiplient. Ils sont généralement moins chers et plus performants que la machine qu'ils imitent. Voici un des premiers portables compatibles IBM, le Compaq Plus.

Le Personal Computer d'IBM a apporté lors de son lancement une certaine respectabilité au micro-ordinateur. Avec la caution d'IBM, les hommes d'affaires se sont trouvés rassurés : les micros devenaient sérieux ! Lorsque les sociétés de logiciels ont commencé à écrire de nombreux programmes pour l'IBM PC, il devint inévitable que beaucoup de fabricants de matériel se mettraient à produire des ordinateurs capables de traiter les programmes écrits pour l'IBM PC, et essaieraient ainsi de prendre part à l'exploitation de cette masse de logiciels disponibles.

Une des premières sociétés de ce type fut sans doute Compact Computer Corporation, société créée spécialement pour produire une machine portable compatible IBM, le Compaq Plus. Le modèle étudié ici est la version 256 K avec double lecteur de disques ; la machine est également disponible avec un seul lecteur ou avec un disque dur de 10 méga-octets.

Paquet compact
Les lignes douces et effilées du boîtier du Compaq se retrouvent sur le devant dépouillé de l'ordinateur et sur la reproduction sage du clavier de l'IBM PC. Des panneaux sur les côtés permettent l'accès aux ports et à l'alimentation, ainsi qu'à un espace où peuvent se loger le câble d'alimentation et la prise de courant — un point important trop souvent négligé !
(Cl. Chris Stevens.)

Porter ce poids

« Portable » est un qualificatif un peu abusif pour une machine de 14 kg. Compaq a d'ailleurs prévu une poignée de vinyl rembourrée qui ménage les doigts pendant le transport !

Comme d'habitude avec les micros portables, le clavier du Compaq bascule sur le devant de la machine. Le Compaq Plus est très robuste et le clavier s'accommode très bien du poids et des manipulations un peu brusques. L'ensemble a très approximativement le même encombrement qu'une machine à coudre. Une fois basculé, le clavier se connecte à la machine par l'intermédiaire d'un épais câble torsadé. Le fabricant affirme que cela permet de poser le clavier de manière confortable. C'est peu convaincant, car, au-delà d'une certaine distance (environ 20 cm), le câble torsadé est trop tendu et rappelle le clavier à lui, ce qui limite la distance entre l'utilisateur et l'écran. Un câble plus fin aurait permis d'éviter cette situation.

Comme on est en droit de l'attendre d'un micro professionnel, les touches du Compaq Plus sont d'une utilisation facile et fiable. La disposition du clavier est identique à celle de l'IBM PC, avec dix touches de fonction à gauche du clavier de machine à écrire, et un bloc de touches numériques à droite. Les utilisateurs habitués à l'IBM PC pourront se servir instantanément du clavier du Compaq sans avoir à apprendre l'emplacement des nouvelles touches. Mais cela ne va pas sans inconvénient : le clavier de l'IBM PC étant loin d'être idéal, les problèmes sont reproduits sur le Compaq Plus ! La touche EXÉCUTION sur l'IBM PC, de même taille que les autres touches, est donc difficilement repérable. Plus ennuyeux : pour ceux qui frappent sans regarder, la touche de bascule du mode majuscule n'est pas à sa place habituelle, sous la touche A. A sa place se trouve la touche /, ce qui veut dire que l'on frappera systématiquement, au début du moins, cette touche pour toutes les frappes de la bascule des majuscules. Ce sont des problèmes qui se reproduisent de manière identique sur le Compaq.

En exploitation MS-DOS (la version Microsoft du PC-DOS), les touches de fonction à la gauche du clavier font office de touches d'édition. Ces touches de fonction changent d'affectation selon le programme d'application exécuté. Avec le BASIC, par exemple, ce sont des touches de saisie d'un mot clé tel que LOAD, SAVE ou LIST. Un système similaire est également utilisé pour les touches de fonction sur les machines MSX.

Le bloc numérique de la droite du clavier a une double fonction. En mode normal, il sert à déplacer le curseur à l'écran, mais en conjonction avec la touche de verrouillage numérique, il fait office de calculateur.

L'écran du moniteur, vert, est de dimensions standards, 17,7 × 13,3 cm, avec une résolution de 80 × 25 caractères. Le texte est très lisible; un bouton à la droite de l'écran permet de régler la luminosité. La configuration standard du Compaq Plus ne comprenant qu'un seul lecteur de disquettes, la disquette système MS-DOS supposera que cela est toujours le cas et demandera que tous les disques soient insérés dans le lecteur A. C'est ennuyeux lors de la copie d'un disque, dans la mesure où l'utilisateur devra continuellement interchanger les disques. MS-DOS peut néanmoins être configuré pour utiliser les deux lecteurs.

Connexions externes

Sous un panneau situé sur le côté droit de la machine se trouvent les ports d'interface. Le Compaq Plus est doté d'un port parallèle de type Centronics pour imprimante, d'une interface RVB, et d'un port HF. Trois cases d'extension sont également prévues pour la connexion de cartes électroniques compatibles IBM. Les extensions principales sont des cartes mémoire, une carte vidéo couleur ou un modem qui permet à l'ordinateur de communiquer *via* le réseau téléphonique.

Un ventilateur sur le côté empêche la machine de chauffer. La circuiterie interne est à l'abri d'un cadre métallique (ce qui contribue au poids important de la machine). Ce boîtier interne n'est pas seulement destiné à protéger des manipulations un peu brusques, mais aussi des interférences radio susceptibles d'interrompre le traitement informatique. Le métal sert également à piéger la chaleur dégagée.

Sur le côté gauche se trouve la prise d'alimentation sous laquelle est placé l'interrupteur marche/arrêt.

Trois manuels sont livrés avec Compaq Plus : un guide de l'utilisateur et des manuels de référence MS-DOS et BASIC. Il s'agit de guides brochés et non en classeur, ce qui est peu fréquent. Les systèmes disques sont rangés avec les manuels dans des étuis en suédine. Des trois manuels, seul le guide de l'utilisateur est clair. La personne qui désire apprendre à utiliser le BASIC ou le MS-DOS devra se tourner vers d'autres livres !

Comme les plus anciens compatibles IBM PC et le PC lui-même, le Compaq Plus utilise le processeur Intel 8088 au lieu du 8086, plus évolué et utilisé par ses jeunes rivaux tel que l'Olivetti M25. Bien que le 8088 soit un processeur 16 bits avec un bus d'adresses à 20 bits, il possède un bus de données sur 8 bits par opposition à celui de 16 bits du 8086. Résultat : le Compaq est bien plus lent à restituer les données que ses concurrents dotés du 8086, et cela même s'il fonctionne à la même vitesse que le PC.

Sur la question absolument vitale de la compatibilité avec les logiciels IBM, le Compaq Plus marque des points. Même un programme notablement difficile comme Lotus 1-2-3 fonctionne sur Compaq. Cela ne doit pas surprendre. Il faut savoir, en effet, que le président de Compaq fait également partie du conseil d'administration de la société Lotus Software ! Notons, parmi les rares logiciels que Compaq ne peut exploiter, le disque de diagnostic IBM ; mais, dans la mesure où ce programme interroge directement le système d'entrée/sortie du BASIC en mémoire morte de l'IBM PC, il n'est pas vraiment étonnant qu'on ne puisse le faire fonctionner sur un quelconque compatible.

Le Compaq Plus est bien l'une des machines compatibles IBM PC parmi les plus fiables du marché, avec une réelle compatibilité logicielle. Cela compte beaucoup pour un achat de l'ordre de 30 000 F. Il n'est en effet pas très agréable de s'apercevoir après coup qu'un programme donné ne fonctionne pas.

Cela dit, n'oublions pas que la machine commence à accuser son âge, et pas seulement à cause du processeur 8088 qui est dépassé. En effet, les portables de la catégorie des plus de 10 kg se font rares. Avec l'amélioration des ordinateurs « porte-documents », l'arrivée prochaine d'un compatible IBM de même taille devient inévitable. Les portables du genre Compaq Plus seront alors devenus inutiles.

Rapport qualité/prix

Les points forts du Compaq Plus sont sa compatibilité IBM et donc l'accès à la formidable mine de logiciels destinés à ce dernier. Nous indiquons ici des ordres de prix d'un système IBM PC satisfaisant aux spécifications de base du Compaq Plus :

CARACTÉRISTIQUE	COMPAQ PLUS	IBM PC
Prix de base	28 000 F	14 000F
Double lecteur 360 K	Standard	3 500 F suppl.
Moniteur couleur haute résolution	Standard*	3 200 F suppl.
Carte sortie moniteur	Standard	2 200 F suppl.
Carte graphique couleur	Standard	2 500 F suppl.
128 K RAM	Standard	1 000 F suppl.
Système d'exploitation sur disque et BASIC	Standard	800 F suppl.
Clavier	Standard	2 300 F suppl.
TOTAL	28 000 F	30 000 F

** Le moniteur standard du Compaq est monochrome haute résolution, mais la machine a en standard une sortie moniteur couleur RVB.*

Chris Stevens

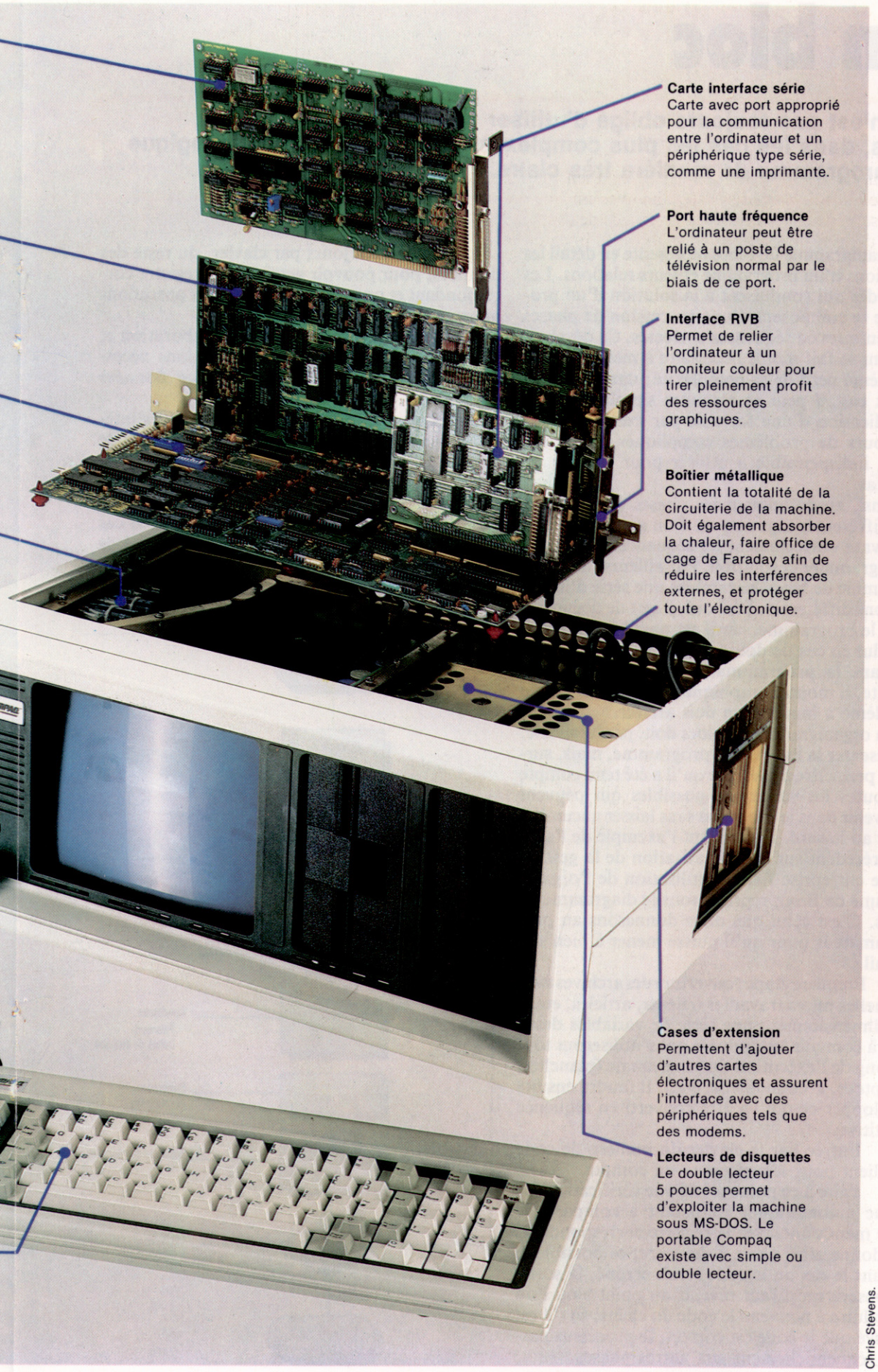
Carte imprimante
Carte avec interface parallèle pour connexion avec une imprimante de type Centronics.

Carte vidéo
Se loge dans une des cases d'extension du Compaq et fournit la circuiterie nécessaire à la gestion d'une télévision ou d'un moniteur externe.

Carte du circuit principal
Comporte les 256 K de RAM, le processeur 8088 et les composants d'entrée/sortie. Il reste des cases libres pour d'autres composants.

Alimentation
Logement de l'alimentation, avec, à côté, un ventilateur destiné à refroidir la machine.

Clavier style IBM
Clavier identique à la version IBM PC. Remarquez les touches de fonction à gauche et le bloc numérique à droite.



Carte interface série
Carte avec port approprié pour la communication entre l'ordinateur et un périphérique type série, comme une imprimante.

Port haute fréquence
L'ordinateur peut être relié à un poste de télévision normal par le biais de ce port.

Interface RVB
Permet de relier l'ordinateur à un moniteur couleur pour tirer pleinement profit des ressources graphiques.

Boîtier métallique
Contient la totalité de la circuiterie de la machine. Doit également absorber la chaleur, faire office de cage de Faraday afin de réduire les interférences externes, et protéger toute l'électronique.

Cases d'extension
Permettent d'ajouter d'autres cartes électroniques et assurent l'interface avec des périphériques tels que des modems.

Lecteurs de disquettes
Le double lecteur 5 pouces permet d'exploiter la machine sous MS-DOS. Le portable Compaq existe avec simple ou double lecteur.

Chris Stevens.

Compaq plus

PRIX
★ ★ ★ ★ ★

DIMENSIONS
480 × 400 × 200 mm.

UC
8088 d'Intel (4,7 MHz).

MEMOIRE
256 K de RAM, extension jusqu'à 640 K.

AFFICHAGE
Texte : 25 lignes de 80 colonnes.
Graphique : 640 × 200 pixels.

INTERFACES
Imprimante parallèle de type Centronics, prise RVB et HF. Autres cases d'extension pour des cartes d'interfaçage.

LANGAGE
BASIC chargé à partir de la disquette système.

CLAVIER
47 touches type machine à écrire, 14 touches de contrôle, 10 touches de fonctions et 10 touches de fonctions de calcul.

DOCUMENTATION
Trois manuels fournis : guide de l'utilisateur, manuels de référence MS-DOS et BASIC. Seul le guide de l'utilisateur est didactique. Les débutants auront intérêt à acheter d'autres manuels pour progresser.

AVANTAGES
Machine très robuste et de compatibilité éprouvée avec l'IBM PC.

INCONVENIENTS
Le poids de la machine la met dans la catégorie « portable ». Le Compaq Plus est en voie d'être dépassé technologiquement.

En bloc

On n'est pas toujours obligé d'utiliser ce type d'organigramme, mais, dans les cas les plus complexes, il met en évidence la logique du programme de manière très claire.

L'organigramme de blocs représente en détail les fonctions d'un programme et leurs relations. Les procédés qui conduisent à la solution d'un problème se composent d'une succession de phases élémentaires en séquences correctes. Ce développement se fait moins nécessaire à mesure que le problème perd de sa complexité; dans de nombreux cas, il peut se résoudre simplement par l'application d'une formule par exemple. Mais au cours de problèmes compliqués, il devient quasi indispensable à utiliser pour aboutir au résultat.

Ainsi, en définissant les phases élémentaires des différentes alternatives d'un problème et en décrivant correctement les séquences, on arrive à programmer la solution. Par ailleurs, un organigramme de blocs hiérarchise une série d'informations différentes. D'abord, il sert de document pour le programme, car il est possible de recourir à lui en cas de doute sur la séquence utilisée pendant la programmation ou après celle-ci, ensuite, il montre graphiquement la solution du problème à laquelle on doit arriver.

Un organigramme de blocs doit, par lui-même, représenter la logique du programme, mais, surtout, permettre de vérifier qu'il a été tenu compte de toutes les variations possibles qui peuvent intervenir dans le processus sans laisser aucun élément au hasard. En prenant l'exemple de l'article précédent sur l'automatisation de la gestion d'une entreprise fictive (utilisation de l'organigramme de flux), représentons le diagramme de blocs. C'est celui que nous donnerons au programmateur pour qu'il puisse mener à bien son travail.

1. Première étape : ouverture des archives avec lesquelles on va travailler (clients, articles, etc.). On dimensionnera une série de variables destinées à contenir les données qui s'utiliseront tout au long de l'exécution du programme (planches, compteur, accumulateur, etc.). Il faudra ensuite développer ce qui va être converti en séquence répétitive.

2. Entrée par le clavier du numéro de code du client pour lequel existe un compte.

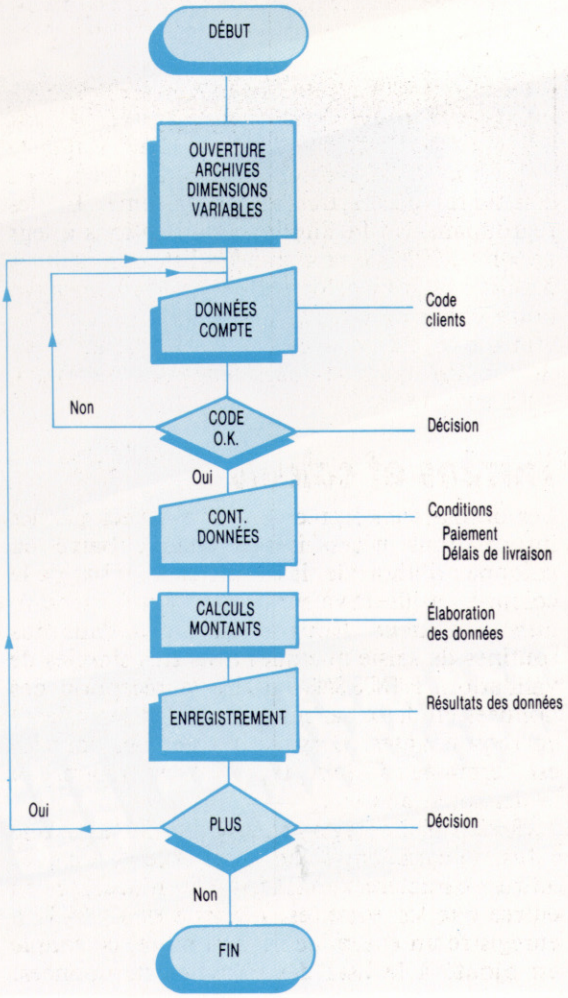
3. Faire à ce point précis une vérification (de forme automatique) consistant à comparer le code mentionné avec le registre correspondant. On donne ainsi accès à deux sorties possibles : *a*) dans le cas où le code serait erroné, la sortie supposera qu'il faut revenir au point b où l'on introduira à nouveau le code du client ; *b*) s'il est prouvé que le code est correct, le processus passera séquentiellement aux autres phases.

4. Entrée, toujours par clavier, du reste des données, pour pouvoir actualiser le registre correspondant et pour les utiliser dans les opérations appropriées.

5. Le dernier point suppose l'élaboration et l'obtention de résultats. Les opérations nécessaires se réaliseront en partant des données disponibles.

6. A ce niveau, les données modifiées s'intègrent dans le registre, et celui-ci s'enregistre de nouveau dans le fichier qui le conserve ainsi.

7. Dernière étape : le processus peut se répéter autant de fois qu'on le veut. Si tel est le cas, la séquence retournera au point b; dans le cas contraire, le programme se termine. L'unité de l'organigramme de blocs paraît tout à fait claire. Un bon programme, dans la plupart des cas, ne devrait pas s'en passer.



Diagrammes hauts de forme

Après le développement de routines et de procédures de gestion des données, ainsi que la création de dessins par la tortue, nous allons utiliser LOGO pour afficher les données de manière visuelle.

Les diagrammes à barres sont une méthode graphique simple pour représenter certains types de données mumériques. Leur but est de permettre à l'utilisateur d'apprécier un ensemble de chiffres au premier coup d'œil, sans avoir à l'examiner en détail. Les programmes graphiques de gestion qui tracent des diagrammes à barres, des « camemberts » et des histogrammes sont très répandus. Ils sont souvent liés à des tableurs, de sorte que les valeurs calculées puissent être représentées.

Nous nous limiterons pour l'instant à la manière de tracer un diagramme à barres avec LOGO. La version présentée ici est destinée au Commodore 64 (voir « Variantes de LOGO » pour les aménagements sur d'autres machines).

Le tracé d'un diagramme à barres se décompose en trois étapes :

a) prendre les données en entrée;

b) trouver la plus grande valeur (il est nécessaire d'étalonner les colonnes pour rester dans l'écran);

c) tracer le diagramme à barres, à l'échelle voulue, et ajouter les titres.

La procédure de niveau le plus élevé s'appelle DIABARRES. INIT initialise un certain nombre de constantes nécessaires au programme. En les regroupant, on facilite les modifications à leur apporter. COULEURS comprend la liste des couleurs à utiliser pour les colonnes — si vous n'aimez pas notre choix de couleurs, modifiez-le! Le programme tel que nous le donnons affiche au maximum quinze colonnes, aussi il ne vous faudra que quinze couleurs.

Entrées et calculs

Les données en entrée sont constituées par les informations nécessaires à chaque barre ou colonne : d'abord le titre à afficher en bas de la colonne, ensuite la valeur pour la barre — c'est-à-dire sa hauteur. Nous avons inclus dans nos routines de saisie quelques tests très simples de validation. PRENDS.SAISIE divise la réception des données en deux parties, les titres et les valeurs correspondantes. Lorsque la saisie des données est terminée, il faut taper FIN en réponse à la demande de titre.

PRENDS.TITRES enregistre les titres. Cette routine refuse le caractère blanc, mais accepte tous les autres caractères. PRENDS.VALEUR n'accepte en entrée que les nombres. Lorsque PRENDS.SAISIE a enregistré un ensemble nom-nombre, ce couple est ajouté à la liste des éléments de données. Représentant autant de barres sur le diagramme, ils doivent être donnés dans l'ordre où on désire les voir figurer, de gauche à droite.

CALCULE utilise PRENDS.MAX pour trouver la plus grande valeur, et se sert de cette valeur pour établir une échelle. Une règle courante veut que la hauteur maximale d'un diagramme à barre soit environ égale aux trois quarts de sa largeur.

Tracer le tableau

TRACE.BARRE calcule la largeur de chaque barre, trace un axe vertical à l'écran, colore les barres et leur donne un nom.

La largeur est calculée par multiples de 8. Cela permet d'éviter certains problèmes dus à l'affichage couleur du Commodore. En effet, dans le mode graphique normal du LOGO, on ne peut avoir deux couleurs différentes dans un même pavé de 8 × 8 pixels. TRACE.AXE trace un axe vertical pour les données. Cela nous donne un moyen très simple d'estimer les valeurs des colonnes.

Il nous faut éloigner le crayon de cet axe vertical pour inscrire la valeur maximale à son extrémité. En ce sens, nous devons déterminer de combien s'écarter pour que le nombre ne le chevauche pas. Cela dépend évidemment du nombre de chiffres dans le nombre. Ce problème est facilement résolu car LOGO considère les nombres comme des mots. Nous pouvons donc utiliser COMPTE pour déterminer la longueur du nombre, et nous servir de cette longueur pour calculer de combien s'écarter de l'axe vertical.

ÉCRIS affiche un message sur l'écran graphique. Cette routine accepte trois données en entrée : les unités de longueur *x* et *y* pour le pas de chaque caractère, et le nom à faire figurer. Elle utilise une primitive, FRAPPECAR, qui frappe le caractère à l'écran graphique sur la position occupée par la tortue.

Camemberts

TRACE.BARRE1 est chargée de tracer les barres. Cette routine prend tour à tour les éléments de données, prend la couleur appropriée, évalue la hauteur pour la barre et transmet l'exécution du travail à REMPLIS. Cette dernière monte et descend dans la colonne-barre et la remplit de couleur. TITRE utilise ÉCRIS pour écrire verticalement les titres vers le bas de l'écran (les titres trop longs s'« enrouleront » sur le bas de l'écran pour réapparaître en haut).

Les « camemberts » sont une autre forme courante de représentation graphique. Voici quelques conseils pour écrire un programme qui les trace :

- les données s'obtiennent exactement de la même manière ;
- la partie calcul comprend la totalisation des valeurs et donc la répartition des parts sur un angle total de 360° ;
- le tracé et le remplissage du camembert sont assez simples, mais vous aurez néanmoins des problèmes sur le Commodore 64 du fait de l'interpénétration des couleurs. Aussi, une bonne idée est d'utiliser le mode double couleur, DOUBLECOULEUR (au lieu de TRACE), dans la procédure qui dessine le tableau. Si vous avez encore des problèmes, ménagez un trou de dix unités au centre du camembert. Les titres sont attribués de la même manière qu'avec le diagramme à barres, même si des problèmes peuvent survenir quand vous positionnez la tortue pour l'écriture.

Variantes de logo

De nombreuses versions LOGO n'ont pas d'équivalents pour FRAPPECAR, ce qui rend l'affichage de caractères sur l'écran graphique particulièrement difficile.
Les nombres pour les couleurs et la taille du tableau devront être modifiés pour chaque machine particulière. Tous ces détails sont réunis dans INIT ; ils sont ainsi traités en une seule fois.

Pour toutes les versions LCSI :

Utilisez TAPE pour AFFICHE1.
Utilisez VIDEP pour VIDE?.
Vous aurez peut-être à utiliser ÉGALP pour le signe =.
DÉTERMINEXY doit être suivi d'une liste.
SI a une syntaxe différente, par exemple :
SI VIDEP :LISTEDONNÉES [STOP]

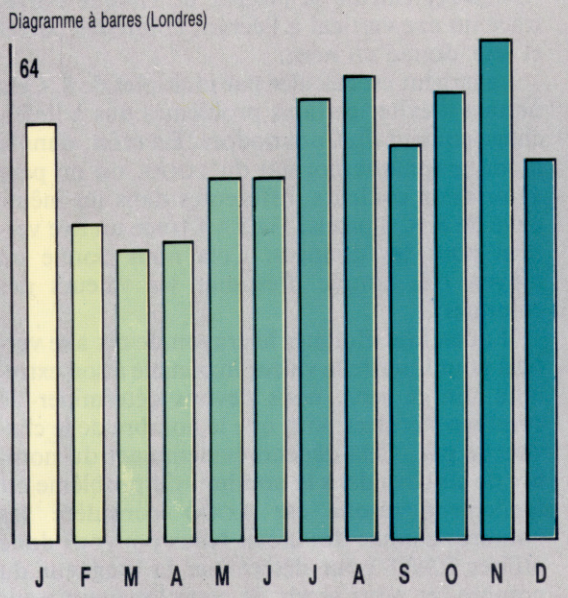
Nom d'un diagramme!

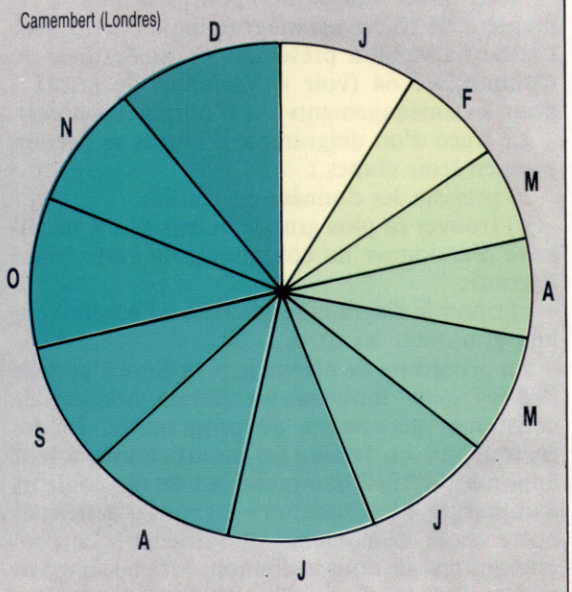
Les diagrammes à barres et les camemberts sont aussi utiles les uns que les autres pour représenter de manière visuelle des données, même si LOGO n'est pas vraiment l'idéal du fait de sa lenteur.

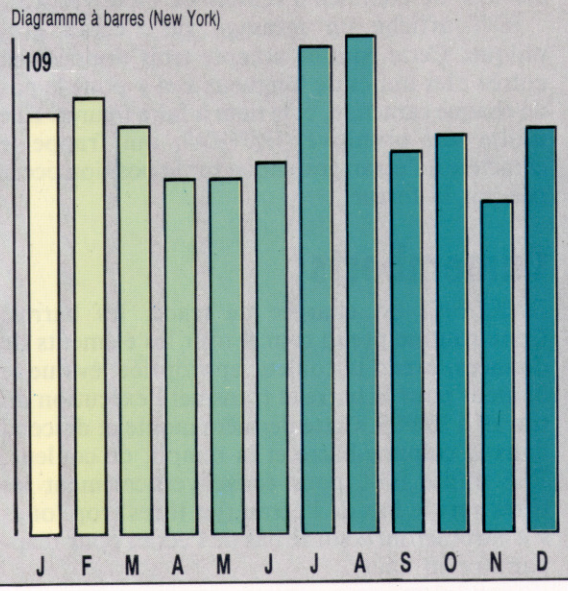
Données en entrée

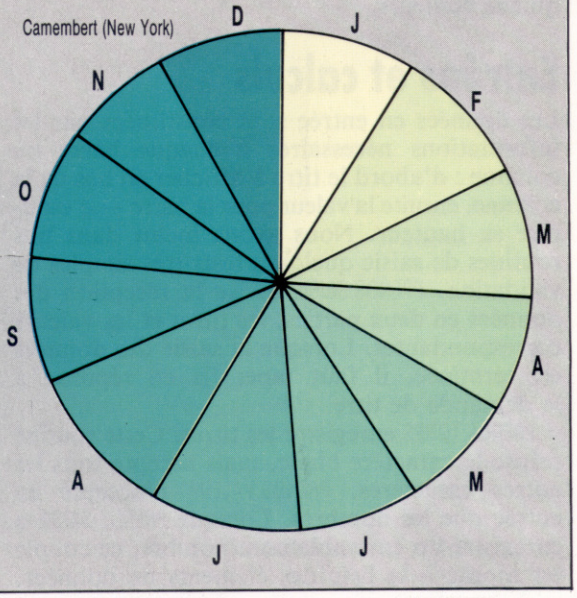
	Londres	New York
JANV.	53	94
FEVR.	40	97
MARS.	37	91
AVR.	38	81
MAI.	46	81
JUIN.	46	84
JUIL.	56	107
AOUT.	59	109
SEPT.	50	86
OCT.	57	89
NOV.	64	76
DEC.	48	91

Données pluviométriques (en millimètres)









Programme de diagramme à barres

```
POUR DIABARRES
  INIT NOTRACE
  PRENDS.SAISIE CALCULE
  TRACE.BARRE
FIN

POUR INIT
  FAIS «COULEURS [0 1 2 5 6 0 1 2 5 6 0 1 2 5 6]
  FAIS «COULFOND 11
  FAIS «DONNÉES []
  FAIS «HAUTEUR 160
  FAIS «LARGEUR 190
  FAIS «XDÉBUT (-104)
  FAIS «YDÉBUT (-40)
FIN

POUR PRENDS.SAISIE
  PRENDS.TITRE
  SI PREMIER :TITRE = «FIN ALORS STOP
  PRENDS.VALEUR
  FAIS «DONNÉES LMETS PHRASE :TITRE
  VALEUR :DONNÉES
  PRENDS.DONNÉES
FIN

POUR PRENDS.TITRE
  AFFICHE «
  AFFICHE [(FRAPPEZ FIN POUR TERMINER)]
  (AFFICHE1 «TITRE: «' ')
  FAIS «DEMANDETITRE
  SI VIDE? :TITRE ALORS PRENDS.TITRE
FIN

POUR PRENDS.VALEUR
  (AFFICHE1 «VALEUR: «' ')
  FAIS «DEMANDEVALEUR
  SI VIDE? :VALEUR ALORS PRENDS.VALEUR
    SINON SI NON NOMBRE?
  PREMIER :VALEUR ALORS PRENDS.VALEUR
FIN

POUR CALCULE
  FAIS «MAX 0
  PRENDS.MAX :DONNÉES
  FAIS «ÉCHELLE :HAUTEUR / :MAX
FIN

POUR PRENDS.MAX :LISTEDONNÉES
  SI VIDE? :LISTEDONNÉES ALORS STOP
  SI DERNIER PREMIER :LISTEDONNÉES > :MAX ALORS
    FAIS «MAX DERNIER PREMIER
  :LISTEDONNÉES
  PRENDS.MAX SAUFPREMIER :LISTEDONNÉES
FIN

POUR TRACE.BARRE
  TRACE FOND :FD
  CACHETORTUE PLEIN ÉCRAN
  FAIS «LARGEUR ( (ENROULE ( ( :LARGE /
    COMPTE :DONNÉES) / 8) ) ★ 8 - 8
  PLUMELÈVE DÉTERMINEXY :XDÉBUT :YDÉBUT
    TRACE.AXE
  TRACE.BARRE1 :COULEURS :DONNÉES
  NOM :DONNÉES
FIN

POUR TRACE.AXE
  PLUMEBAISSE
  DÉTERMINEY YCOOR + :HAUTEUR + 10
  DÉTERMINEX XCOOR + 4
  DÉTERMINEX XCOOR - 8
  PLUMELÈVE
  DÉTERMINEX XCOOR - (8 ★ COMPTE :MAX)
  ÉCRIS 8 0 :MAX
  DÉTERMINEX XCOOR + 4
  DÉTERMINEY :DÉBUT
FIN

POUR ÉCRIS :XINC :YINC :NOM
  SI VIDE? :NOM ALORS STOP
  FRAPPECAR PREMIER :NOM
  DÉTERMINEXY XCOOR + :XINC YCOOR + :YINC
  ÉCRIS :XINC :YINC SAUFPREMIER :NOM
FIN

POUR TRACE.BARRE1 :COULEURSPLUME
  :LISTEDONNÉES
  SI VIDE? :LISTEDONNÉES ALORS STOP
  DÉTERMINEX XCOOR + 8
  COULEURPLUME PREMIER :COULEURSPLUME
  PLUMEBAISSE
  REMPLIS ( (DERNIER PREMIER :LISTEDONNÉES) ★
  :ÉCHELLE)
    LARGEUR
  PLUMELÈVE
  TRACE.BARRE1 SAUFPREMIER :COULEURSPLUME
    SAUFPREMIER :
    LISTEDONNÉES
FIN

POUR REMPLIS :LONGUEUR :LARGEUR
  SI LARGEUR = 0 ALORS STOP
  AVANCE :LONGUEUR DROITE 90
  AVANCE 1 GAUCHE 90
  DEMITOUR :LONGUEUR DROITE 90
  AVANCE 1 GAUCHE 90
  REMPLIS :LONGUEUR :LARGEUR - 2
FIN

POUR TITRE :LISTEDONNÉES
  TITRE1 (:XDÉBUT + 8 + :LARGEUR / 2) (:YDÉBUT - 10)
  :LISTEDONNÉES
FIN

POUR TITRE1 :X :Y :LISTEDONNÉES
  SI VIDE? :LISTEDONNÉES ALORS STOP
  DÉTERMINEXY :X :Y
  ÉCRIS0 (- 10) PREMIER PREMIER :LISTEDONNÉES
  TITRE1 (:X + :LARGEUR + 8) :Y SAUFPREMIER
    :LISTEDONNÉES
FIN
```

Micro-interrupteur

La première phase du montage de notre robot est terminée, et nous l'avons testée en écrivant un court programme. Nous allons à présent monter quatre capteurs à micro-interrupteurs sur le robot et écrire un programme simple pour tester leur fonctionnement.

La commande bidirectionnelle des moteurs pas à pas nécessite quatre des huit lignes de données du port utilisateur disponibles. Il reste quatre lignes pour véhiculer l'information entre les capteurs et l'ordinateur. Pour rendre le fonctionnement de notre robot plus souple, nous utiliserons un « réseau » qui permettra une permutation des connexions des capteurs aux quatre lignes d'entrée disponibles. Pour le moment, nous connecterons quatre capteurs à micro-interrupteurs, puis nous installerons deux capteurs lumineux. Afin de pouvoir sélectionner toute combinaison de ces capteurs — par exemple deux capteurs à micro-interrupteurs et deux capteurs lumineux — nous connecterons chaque capteur à une prise sur le couvercle du robot. Quatre prises seront également connectées aux lignes de données — D_4 à D_7 — sur la fiche D. Nous pouvons donc connecter le capteur approprié à l'une des quatre lignes de données en utilisant un petit circuit.

Pour tester la construction et le montage, nous pouvons écrire un programme très simple qui examine les quatre bits supérieurs du registre de données et qui affiche les valeurs décimales des bits mis en bas du registre.

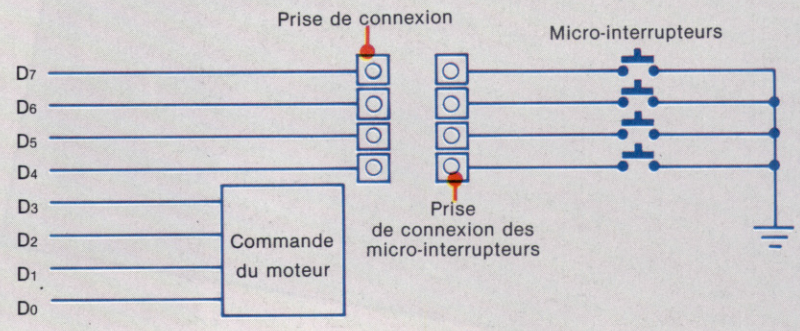
Installation des micro-interrupteurs

La liste des pièces nécessaires est offerte par divers distributeurs. Faites d'abord les découpes nécessaires sur le couvercle du boîtier du robot. Les huit découpes doivent être assez grandes pour y introduire les plaques de connexion sous chaque micro-interrupteur. Les trous de 10,5 mm sont destinés aux prises de connexion — les quatre étant montées en ligne près de la fiche D. Les micro-interrupteurs doivent être légèrement adaptés pour ce projet. Le long levier du micro-interrupteur doit être plié avec précaution à l'aide d'une paire de pinces afin d'obtenir un angle droit. On doit veiller à ce que la courbure ne se trouve pas trop près du boîtier du micro-interrupteur, afin que le levier puisse fermer celui-ci lorsqu'il est monté sur le couvercle. Deux plaques de connexions apparaissent en saillie au dos du micro-interrupteur. Celle du dessus est le connecteur N.F. « normalement fermé ». Comme il ne sera pas utilisé dans notre projet, vous pouvez le retirer, soit en le cassant soit en le sciant. Le connecteur inférieur, « normalement ouvert », est utilisé, et doit être plié à angle droit près du boîtier de l'interrupteur. Cela assure son passage, avec le connecteur commun, par les ouvertures qui lui sont destinées dans le couvercle du boîtier du robot. Lorsque ces modifications ont été effectuées sur chacun des quatre micro-interrupteurs, ils peuvent être montés à chaque coin du couvercle (voir illustrations). Les interrupteurs doivent être montés de façon que les leviers d'activation dépassent à l'avant et à l'arrière du boîtier du robot pour être collés en place, à l'aide d'un adhésif adéquat. Les colles au cyanoacrylate sont probablement les plus indiquées.

```
10 REM **** EST CAPTEUR BBC ****
20 MODE 7:OP=-1:DDR=&FE62:DATREG=&FE60:?DDR=15
30 PE=240-(?DATREG AND 240):IF PE=OP THEN 30
40 CLS:PRINT PE:OP=PE:GOTO 30
```

```
10 REM **** TEST CAPTEUR CBM ****
20 OP=-1:DDR56579:DATREG=56577:POKE DDR,15
30 PE=240-(PEEK(DATREG) AND 240):IF PE=OP THEN 30
40 PRINT CHR$(147):PRINT PE:OP=PE:GOTO 30
```

Circuit micro-interrupteur

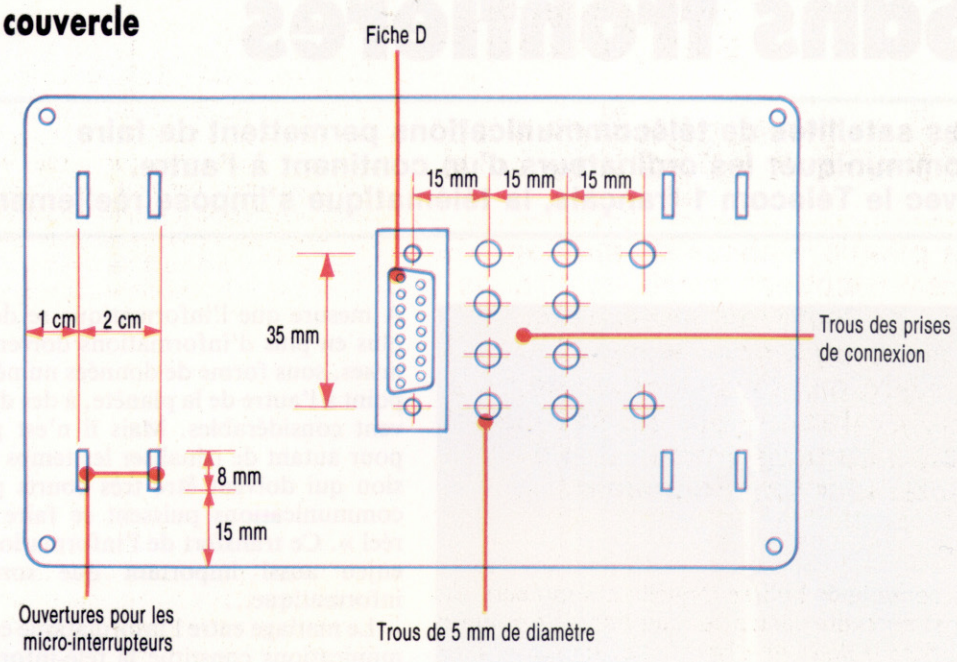


Circuit du micro-interrupteur

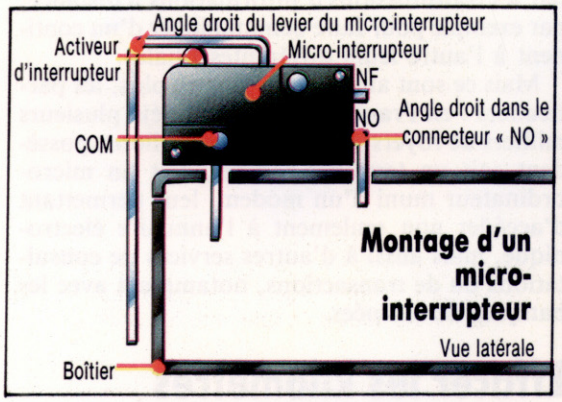
Le circuit qui connecte les capteurs à micro-interrupteur au système du robot est très simple. Les lignes de données D_4 à D_7 sont connectées aux quatre prises montées sur le couvercle du robot; les lignes de données D_0 à D_3 servent à commander le moteur. L'un des côtés de chaque micro-interrupteur est connecté à un groupe similaire de quatre prises; l'autre côté est connecté à une prise de terre commune. Si les quatre micro-interrupteurs sont nécessaires, ils peuvent être connectés aux lignes de données par quatre cordons de connexion. Si les quatre bits supérieurs du registre de données du port utilisateur sont mis en mode entrée, ils sont normalement maintenus au niveau haut (à 1). Fermer tout micro-interrupteur relié au système connecte la ligne de données associée à la prise de terre, ce qui fait passer le bit correspondant du registre de données au niveau bas (à zéro).

Trous du couvercle

Vu du dessus



Kevin Jones



Liste des pièces

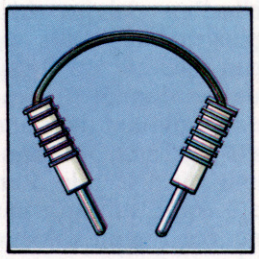
Nombre	Élément	Source
8	fiches 2 mm	HF38R
4	prises rouges 2 mm	HF44X
6	prises bleues 2 mm	HF44X

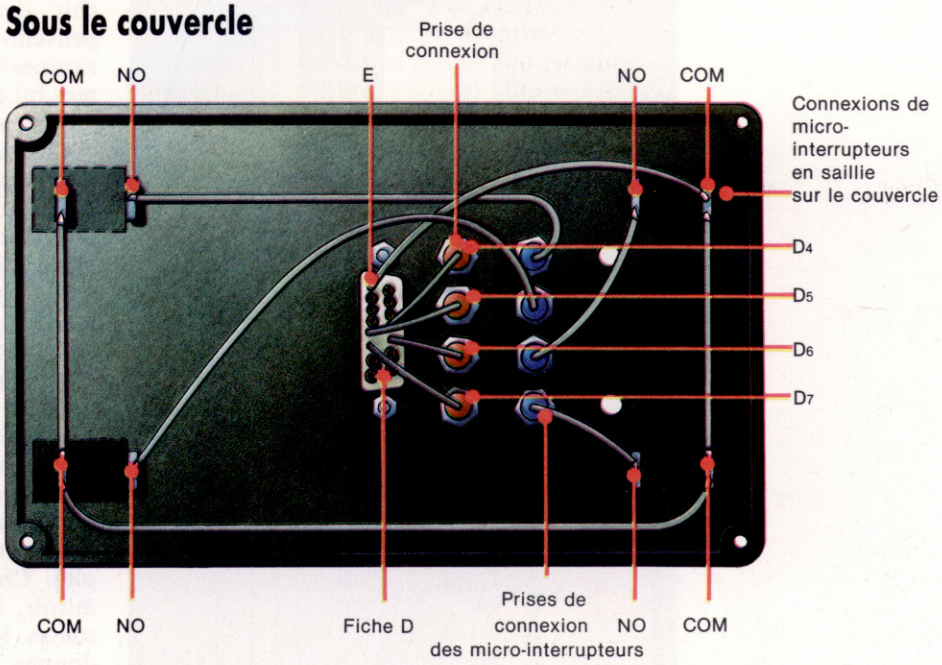
Divers

4 micro-interrupteurs	E22
1 m de câble-ruban 4 voies	

Sous le couvercle

Les micro-interrupteurs étant montés, retournez le couvercle. Soudez de courts fils gainés à chacune des prises de connexion rouges et connectez celles-ci aux broches associées de la fiche D. Les quatre connecteurs COM doivent être connectés entre eux et à la prise de terre de la fiche D dans un circuit annulaire. Chaque connecteur NO doit être connecté à la prise bleue appropriée.

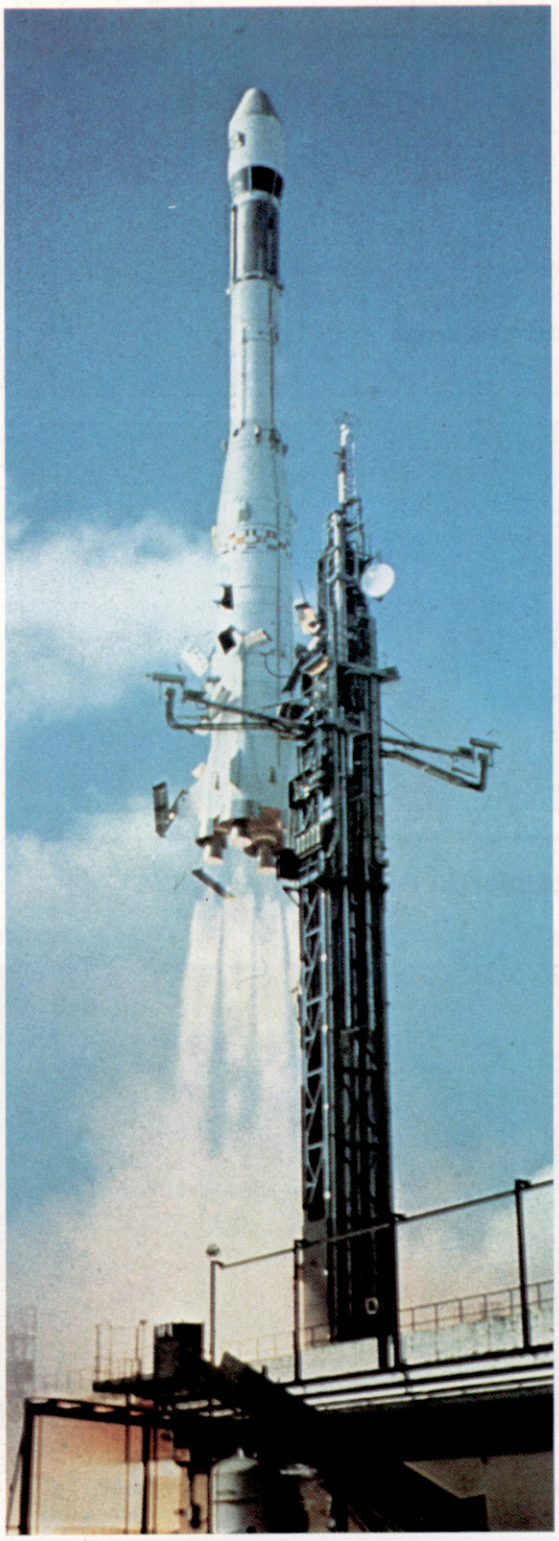




Kevin Jones

Sans frontières

Les satellites de télécommunications permettent de faire communiquer les ordinateurs d'un continent à l'autre. Avec le Télécom 1 français, la télématique s'impose réellement.



Un succès de plus pour la fusée Ariane : le 4 août 1984, le modèle Ariane 3 mettait sur orbite le premier satellite de télécommunications français (le premier satellite national de ce genre lancé par un pays d'Europe), Télécom 1. (Doc. Télécom 1.)

A mesure que l'informatique se développe, de plus en plus d'informations doivent être transmises, sous forme de données numériques, d'un point à l'autre de la planète, à des distances souvent considérables. Mais il n'est pas question pour autant de pénaliser les temps de transmission qui doivent être très courts pour que les communications puissent se faire en « temps réel ». Ce transfert de l'information devient un enjeu aussi important que son traitement informatique...

Le mariage entre l'informatique et les télécommunications constitue la télé-informatique. Ce sont surtout les entreprises qui sont concernées par les transmissions d'informations à distance, par exemple pour faire communiquer d'un continent à l'autre leurs différentes filiales.

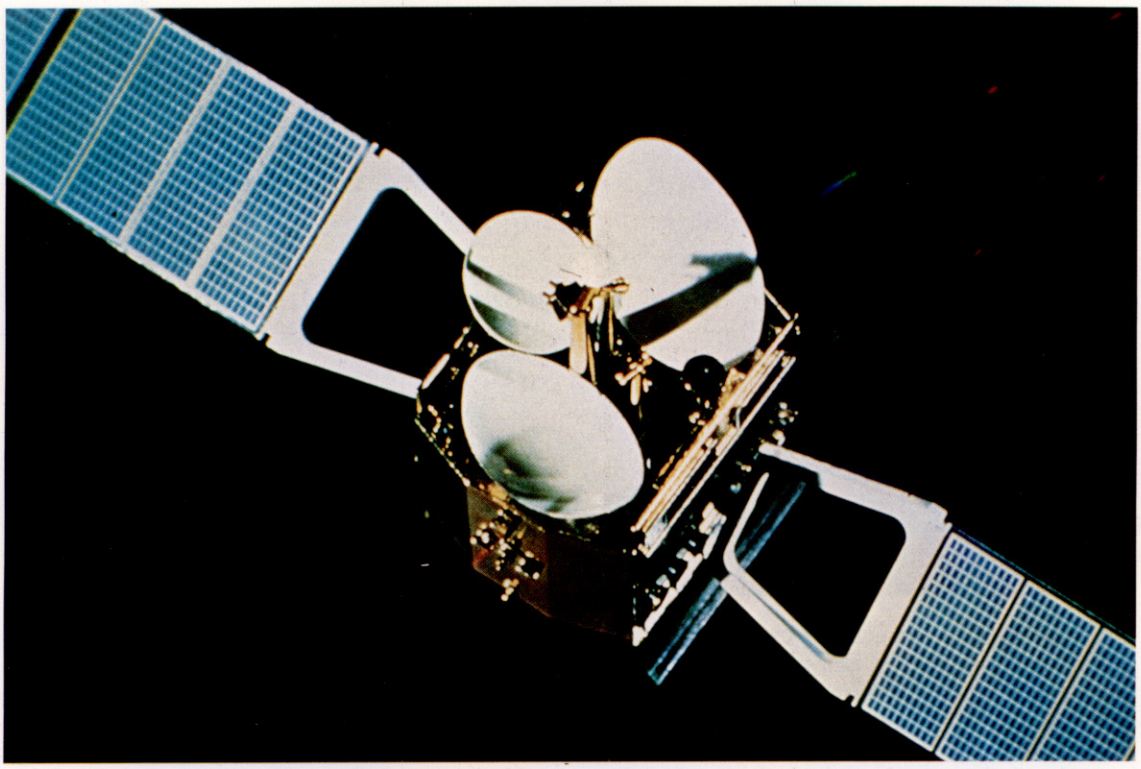
Mais ce sont aussi, et de plus en plus, les particuliers : en France aujourd'hui, déjà plusieurs milliers de foyers, et bientôt des millions, possèdent soit un terminal Minitel, soit un micro-ordinateur muni d'un modem, leur permettant d'accéder non seulement à l'annuaire électronique, mais aussi à d'autres services de consultations ou de transactions, notamment avec les banques de données.

Effacer les kilomètres

Pour arriver à ces services, il faut parfois franchir des océans. Comment les Européens peuvent-ils accéder aux banques de données américaines ? Est-il possible d'échanger des programmes ou des informations, *via* un modem, d'un continent à l'autre ?

Ces questions sont devenues réalité grâce aux satellites de télécommunications, et notamment à Télécom 1, le premier d'une famille de quinze satellites développés par Matra-Espace à Toulouse, complétée par la plate-forme Eurostar, celle-ci étant développée en commun avec British Aerospace, l'entreprise équivalente de la SNIAS en France.

La fonction d'un satellite de télécommunications est très similaire à celle d'un relais hertzien. Un satellite comporte plusieurs répéteurs ou « transpondeurs », chacun recevant un signal dans une bande de fréquences donnée (6 gigahertz, par exemple), et le réamplifiant dans une autre bande de fréquences (4 gigahertz, par exemple). Chaque répéteur joue ainsi le rôle d'un miroir, le faisceau renvoyé étant utilisable par toutes les stations d'une zone géographique donnée.



Télécom 1 est le premier d'une famille de quinze satellites de télécommunications mis au point par la société Matra-Espace. Il a été placé, comme tous ses « confrères », sur une orbite géostationnaire à 36 000 km d'altitude. (Doc. Télécom 1.)

Ariane 3 lance Télécom 1

En s'envolant du centre spatial guyanais de Kourou, le 4 août 1984, la fusée européenne Ariane 3 remplissait une double mission : la mise sur orbite de deux satellites de télécommunications, ECS-2 pour l'Europe et Télécom 1 pour la France. Par télécommandes émises depuis Toulouse, Télécom 1 a été placé sur son orbite géostationnaire située à une altitude de 36 000 km. A cette distance, qui est celle de tous les satellites géostationnaires, la période de rotation autour de la terre est de 24 heures exactement, de sorte que, pour un observateur terrestre, le satellite paraît immobile.

Décidé en 1979 par la Direction générale des télécommunications (D.G.T.), Télécom 1 complète la gamme des services numériques intégrés qui sont déjà à la disposition des entreprises ; c'est un système de communication ouvert sur l'avenir : en effet, sa philosophie, les interfaces utilisées le rendent cohérent avec tout ce qui se fera par la suite.

Avec Télécom 1, la France est le premier pays d'Europe à posséder son propre satellite de télécommunications. Le système répond aux besoins résultant du développement de l'informatique et de la vidéocommunication. Il a un triple objectif :

- fournir des moyens de télécommunications numériques dans une grande gamme de débits pour les communications entre les établissements d'une entreprise ;
- fournir un service de vidéotransmission pour la diffusion de programmes de télévision spécifiques à usage professionnel, éducatif ou culturel ;
- assurer les liaisons téléphoniques et télévisuelles entre la métropole et les DOM-TOM.

Outre Télécom 1, le système comprendra un deuxième satellite de secours en orbite, Télécom 1B, ainsi qu'un troisième au sol, Télécom 1C, qui ne sera lancé qu'en cas de défaillance du satellite principal et du satellite de secours.

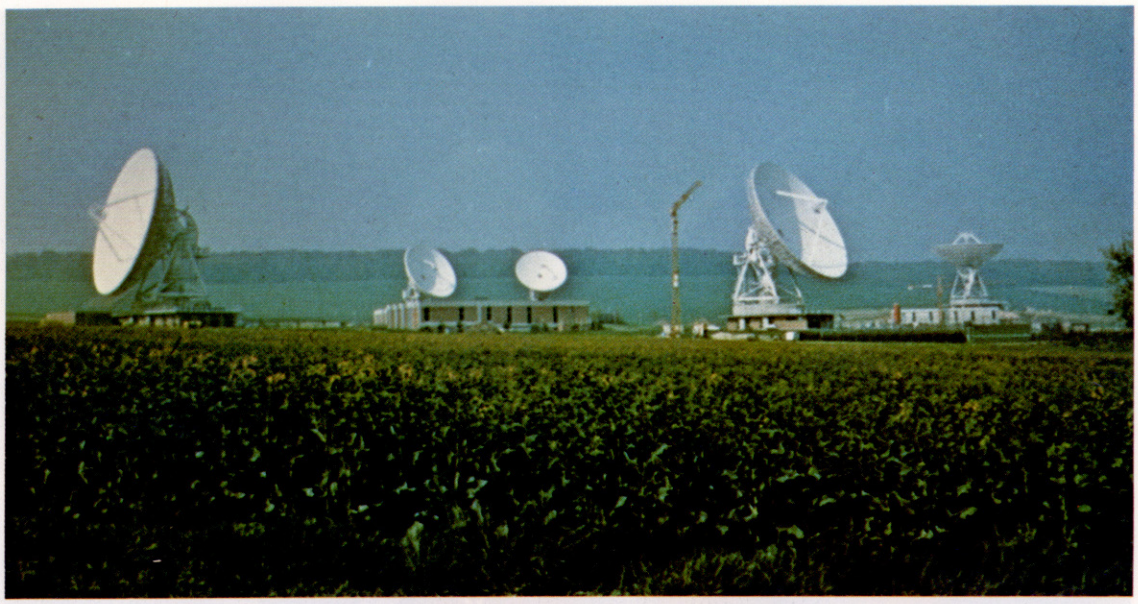
La souplesse du « tout numérique »

Sur Télécom 1, toute l'information est relayée sous forme numérique : fichiers images, paroles utilisant le langage des informaticiens.

D'une capacité de débit total de 150 Mbits/s, Télécom 1 offre aux utilisateurs une vaste gamme de débits : de 2 400 bits/s à 2 Mbits (30 canaux à 64 Kbits/s). Le canal de base est à 64 Kbits/s. Différentes configurations de réseau sont mises en place pour satisfaire à la plage des débits proposés. Cinq répéteurs offrent chacun un débit de 25 Mbits/s pour la transmission des données, et un sixième répéteur assure la transmission d'images. La gestion de cette capacité est assurée par une station de référence qui l'affecte en fonction des demandes, synchronise les émissions de chaque station, fournit des données de trafic et de tarification. Ainsi, par exemple, d'un bout à l'autre de la France, une liaison à 64 Kbits/s utilisée une heure coûte 432 F.

Les entreprises peuvent utiliser les services de Télécom 1 soit pour les liaisons intra-entreprises, soit pour les liaisons avec d'autres organismes. La connexion des équipements des utilisateurs au système Télécom 1 s'effectue par l'intermédiaire de stations terrestres équipées de petites antennes (3 m de diamètre environ).

L'accès au satellite utilise la technique d'« accès multiple à répartition dans le temps » (A.M.R.T.) :



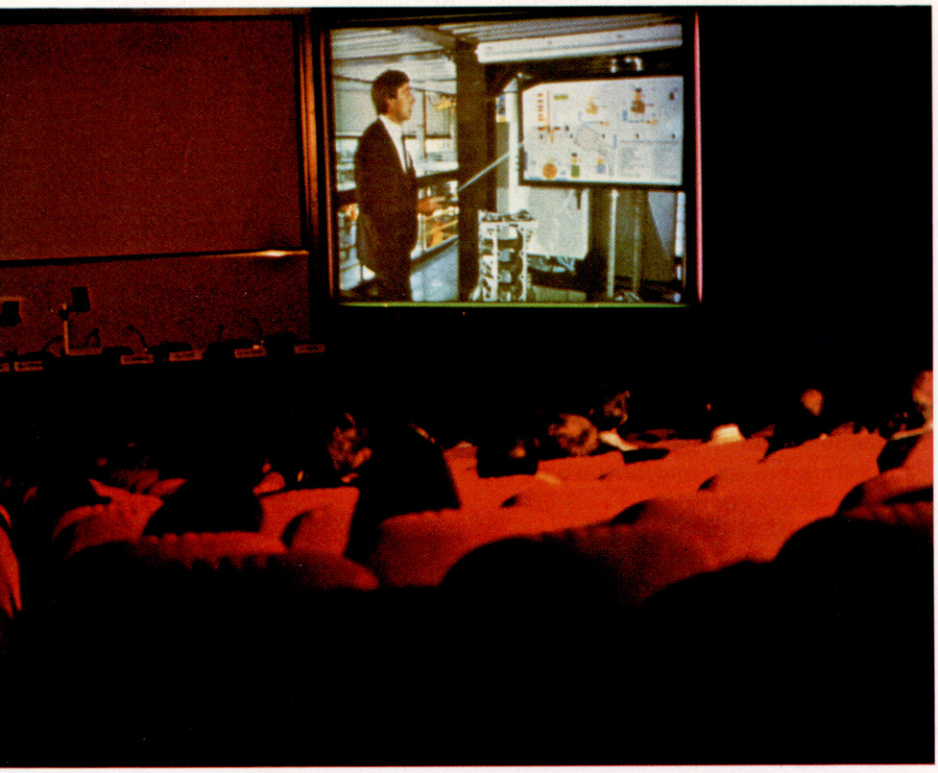
Pour profiter des services de Télécom 1, les clients doivent connecter leurs équipements d'émission et de réception à des stations terrestres munies d'antennes de trois mètres de diamètre environ. (Doc. Télécom 1.)

le temps d'accès « à la carte » est réparti en tranches horaires affectées aux utilisateurs en fonction de leur demande. Ainsi, une entreprise accédera au satellite uniquement pendant le temps nécessaire à ses besoins, soit appel par appel, soit par réservation pendant des tranches horaires. Bientôt, il sera aussi possible d'établir des liaisons en composant le numéro de votre correspondant. Enfin, le choix entre des liaisons point à point, ou une diffusion multipoints, permet de répondre à l'ensemble des problèmes posés.

Avec Télécom 1 naissent des services informatiques véritablement distribués; parmi les applications évidentes : la mise à jour de logiciels, l'édition simultanée d'une documentation et, bien sûr, le transfert de fichiers. Là où, pour des banques, des assurances, des centres de recherche, des nuits entières de transmission suffisent à peine, la puissance du satellite permet d'effectuer des transferts de masse en quelques dizaines de minutes.

Une transmission erronée pourra être renouvelée dans des délais compatibles avec la marche normale d'un service. La sécurité va plus loin avec Télécom 1. Réseau reconfigurable par nature, il permet à un centre informatique vital d'être secouru en cas de défaillance. De n'importe quel autre site, dans la zone de couverture européenne du satellite, le centre de secours peut être reconfiguré rapidement, selon une procédure étudiée au préalable.

Cinq répéteurs offrent chacun un débit de 25 Mbits/s pour la transmission de données et un sixième assure la transmission des images. (Doc. Télécom 1.)



La sécurité d'un réseau reconfigurable

Télécom 1 permet d'ajouter à la dynamique de l'entreprise par toutes les applications en voie de développement. Vis-à-vis des demandes exprimées par les différents services, le responsable informatique aura sans doute à jouer un rôle de conseil et de maître d'œuvre.

Par exemple, quand des établissements dispersés sont impliqués dans un projet commun, moyens de calculs et de C.A.O. (conception assistée par ordinateur) peuvent être reliés par Télécom 1. Les donnés images les plus élaborées peuvent être transmises en quelques secondes.

Les possibilités s'étendent en ce domaine aux principales applications télématiques : télé-écriture, télécopie, transfert de plan en haute définition... Du simple croquis, modifié à des centaines de kilomètres, jusqu'aux plans définitifs, un projet peut être entièrement suivi à distance.

D'autres applications mettent en jeu un autre type d'informations : l'audioconférence, la visioconférence. Limitant la fréquence des déplacements, le temps perdu, elles prolongent les effets des contacts personnels, commerciaux ou administratifs.

Page manquante
(publicité)

Page manquante
(publicité)